

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

BAKALÁŘSKÁ PRÁCE

**Rozšíření vývojového systému
pro mikropočítače řady 8051**

**Expansion development system
for microcontroller series 8051**

Autor:	Lukáš Traspe
Vedoucí bakalářské práce:	Ing. Josef Grosman
Konzultant:	Ing. Tomáš Martinec

V Liberci 3. 5. 2007

Lukáš Traspe

Zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé BP a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 3.5.2007

Poděkování

Touto cestou bych chtěl poděkovat vedoucímu a konzultantovi své bakalářské práce Ing. Jiřímu Grosmanovi a Ing. Tomáši Martincovi, především za odborné a trpělivé vedení během celého projektu a za pomoc při zpracování této bakalářské práce.

Dále bych chtěl poděkovat celé své rodině za všestrannou podporu při mém vysokoškolském studiu.

Technické univerzitě v Liberci za poskytnutí finančních prostředků na součástky a výrobu plošných spojů.

Firmě TECHNOLINE s.r.o jak za poskytnutí prostor a zapůjčení materiálu k sestavení a oživení výukových desek, tak i za zapůjčený software a počítač.

Mé díky patří také všem učitelům a profesorům, kteří mě při studiu vedli a podporovali a díky nimž jsem tento projekt mohl dokončit.

Abstrakt

Rozšíření vývojového systému pro mikropočítače řady 8051

Cílem této bakalářské práce bylo vytvořit pomůcky pro výuku v oblasti aplikací monolitických mikropočítačů řady 8051. Pomocí této práce se studenti seznámí s programováním, připojováním a navrhováním obvodů k jednočipovému mikroprocesoru AT89C51ED2, který je umístěn na vývojové desce ED2.

Byly navrženy a sestaveny tři výukové desky. Důležitým úkolem této práce bylo vytvořit ukázkové aplikace, které budou demonstrovány na sestavených přípravcích. S tímto úkolem souviselo i navržení ukázkových zadání příkladů pro výuku na sestavených deskách. Pro každou sestavenou desku byly navrženy a odladěny funkční vzorky aplikací včetně ukázkových zadání. Demonstrační příklady slouží zároveň jako kontrola správné funkčnosti desek.

Klíčová slova: Jednočipový mikroprocesor, LCD, krokový motorek, klávesnice.

Abstract

Expansion development system through microcontroller series 8051

The aim of this work was creating instruments for education of monolithic microprocessor series 8051 application. Students will be briefed by programming, connecting and designing of circuitry to single-chip microprocessor AT89C51ED2 which is situated in development board ED2.

There was designed and constructed three educational boards. Important task of this work is creating of demonstration application, which will be shown at constructed preparatives. With this task connected designing of demonstration submission of example for education at constructed boards. For each constructed board was designed functional sample of application also with demonstration submission. Demonstrative examples are also a control of boards right functionality.

Key words: Singlechip microcontroller, LCD, step motor, keyboard

Obsah

Zadání	2
Prohlášení	3
Poděkování	5
Abstrakt	6
Seznam obrázků	10
Seznam zkratk	11
Úvod	15
1. VÝVOJOVÁ DESKA ED2.....	16
1.1 ZÁKLADNÍ VLASTNOSTI A PARAMETRY PROCESORU AT89C51ED2	17
1.2 KOMUNIKACE DESKY ED2 S PROGRAMOVÝM VYBAVENÍM.....	18
1.2.1 uScope.....	18
1.2.2 FLIP	18
1.3 SHRUTÍ TECHNICKÝCH PARAMETRŮ DESKY ED2	19
1.4 MOŽNOSTI ROZŠÍŘENÍ DESKY ED2	19
1.4.1 Expandér - specializované obvody (8255,8155)	20
2. VLASTNOSTI MIKROPROCESORU AT89C51ED2 A ŘADY 8051.....	22
2.1 ČASOVÁNÍ A NULOVÁNÍ MIKROPROCESORU.....	22
2.2 ORGANIZACE PAMĚTI.....	23
2.2.1 Přímé a nepřímé adresování	24
2.3 ČÍTAČE A ČASOVAČE	24
2.4 PŘERUŠOVACÍ SYSTÉM.....	25
2.5 PORTY.....	26
3. PRVNÍ VÝUKOVÁ DESKA VDT1	28
3.1 PŘIPOJOVÁNÍ SOUČÁSTEK NA DESCE VDT1	28
3.1.1 L-Bargraf 28	
3.1.1.1 Připojování L-Bargrafu	29
3.1.1.2 Programování L-Bargrafu	30
3.1.2 Maticová klávesnice na desce VDT1	30
3.1.2.1 Připojování klávesnice	30
3.1.2.2 Programování klávesnice	31
3.1.3 LCD-Displej na desce VDT1	31
3.1.3.1 Připojování LCD-displeje	32
3.1.3.2 Znaková sada LCD-displeje	33
3.1.3.3 Programování LCD-displeje	34
4. DRUHÁ VÝUKOVÁ DESKA VDT2	36
4.1 SEZNÁMENÍ S PŘIPOJOVÁNÍM SOUČÁSTEK K DESCE VDT2.....	36

4.1.1	Segmentové displeje na desce VDT2.....	36
4.1.1.1	Připojování segmentového displeje.....	36
4.1.1.2	Programování segmentového displeje.....	39
4.1.2	Tlačítka na desce VDT2.....	39
4.1.2.1	Připojování tlačítek	39
4.1.3	Reproduktor na desce VDT2.....	41
4.1.3.1	Připojování reproduktoru	41
4.1.3.2	Programování reproduktoru.	43
4.1.4	Obvod reálného času RTC (real time clock)	43
4.1.4.1	Připojování RTC	44
4.1.4.2	Programování RTC	44
4.1.4.3	Funkce Rozhraní SPI	46
5.	TŘETÍ VÝUKOVÁ DESKA VDT3.....	49
5.1	TEORIE O KROKOVÝCH MOTORECH	49
5.1.1	Výhody a nevýhody mezi unipolárním a bipolárním zapojováním.....	51
5.2	H-MŮSTEK	51
5.2.1	Připojování L293D.....	52
5.2.2	Řízení obvodu L293D	53
5.2.3	PWM (Pulse Width Modulation)	54
5.3	OSTATNÍ SOUČÁSTKY NA DESCE VDT3	55
5.3.1	Segmentový displej na desce VDT3	55
5.3.2	Tlačítka na desce VDT3.....	57
6.	NÁVRH, VÝROBA, OSAZENÍ A OŽIVENÍ VÝUKOVÝCH DESEK.....	58
6.1	PROGRAMOVÁNÍ DESKY VDT1.....	58
6.1.1	Fragmenty zdrojového kódu k desce VDT1.....	58
6.1.1.1	Zdrojový kód pro klávesnici	58
6.1.1.2	Zdrojový kód pro LCD.....	59
6.2	PROGRAMOVÁNÍ DESKY VDT2.....	61
6.2.1	Fragmenty kódu softwarové komunikace SPI.....	61
6.3	PROGRAMOVÁNÍ DESKY VDT3.....	62
6.3.1	Fragment kódu pro roztočení motorku vlevo.	63
6.4	ZADÁNÍ KE CVIČENÍM	64
6.4.1	Deska VDT1	64
6.4.2	Deska VDT2	64
6.4.3	Deska VDT3	64
	Závěr.....	66
	Použitá Literatura.....	67
	Přílohy	68
	Příloha č. 1 - schéma desky ED2	69

Příloha č. 2 - elektrotechnická dokumentace k VDT1, návod na cvičení	71
Příloha č. 3 - elektrotechnická dokumentace k VDT2, úprava el. dokumentace, návod ke cvičení	85
Příloha č. 4 - elektrotechnická dokumentace k VDT3, úprava el. dokumentace, návod ke cvičení.....	100
Příloha č. 5 - Návod ke cvičení k programu uScope a FLIP	113

Seznam obrázků

Obr.1:	Skutečné zobrazení desky ED2.....	17
Obr.2:	Taktování vnitřním oscilátorem – připojení krystalu.....	22
Obr.3:	Resetovací obvod v aktivní úrovni. RST=log1	23
Obr.4:	Struktura paměťového prostoru programu.....	23
Obr.5:	Přirozená priorita u AT89C51ED2	26
Obr.6:	Deska VDT1	28
Obr.7:	Schématické znázornění zapojení L-Bargrafu	29
Obr.8:	Schématické zapojení součástky L-Bargraf.....	29
Obr.9:	Schéma maticově snímané klávesnice	31
Obr.10:	Funkce vývodů LCD-displeje.....	32
Obr.11:	Znaková sada LCD-displeje.....	33
Obr.12:	Předdefinované funkce LCD-displeje.....	35
Obr.13:	Význam jednotlivých bitů v bytech instrukcí LCD-displeje.....	35
Obr.14:	Deska VDT2	36
Obr.15:	Zapojení jednoho sedmissegmentového displeje	37
Obr.16:	Popis jednotlivých segmentů	39
Obr.17:	Zdvíhací odpory s připojenými tlačítky na portu P0.....	40
Obr.18:	Zapojení reproduktoru	42
Obr.19:	Obr. Zápis dat do paměti obvodu RTC.....	44
Obr.20:	Čtení dat z paměti obvodu RTC	44
Obr.21:	Funkce obvodu RTC s přiřazenými adresami	45
Obr.22:	Kontrolní registr.....	46
Obr.23:	Stavový registr	46
Obr.24:	Koncepce systému se sběrnici SPI.....	47
Obr.25:	Propojení obvodů Master a Slave	48
Obr.26:	Význam parametrů CPOL a CPHA	48
Obr.27:	Deska VDT3	49
Obr.28:	Řez krokovým motorem s permanentním magnetem	50
Obr.29:	Povolení sepnutí výkonového obvodu	52
Obr.30:	Zapojení H-můstku	53
Obr.31:	Řízení H-můstku dvěma průběhy v protifázi	54
Obr.32:	Pulzně šířková modulace	55
Obr.33:	Zapojení segmentové displeje na desce VDT3	56

Seznam zkratek

+5V	Napájecí napětí o hodnotě 5V
+UCC	Napájecí napětí
3-WIRE	Typ komunikační sběrnice
A/D	Analogově digitální převodník (převádí hodnoty analogového napětí na bitové kombinace)
AIE0	Bit v kontrolním registru umožňující povolení přerušení alarmu 0 u RTC
AIE1	Bit v kontrolním registru umožňující povolení přerušení alarmu 1 u RTC
ALE	Signál z procesoru - používá se k zachycení adresového bytu do externího klopného obvodu
Anoda	Představuje elektrodu s kladným napětím
CAAL	Volání podprogramu
CE	Potvrzující bit u komunikace SPI
CMOS	Typ zapojení tranzistorů využívá oba typy tranzistorů NMOS i PMOS
CNC	System, který obsahuje počítač nebo mikroprocesor jako svou integrální část
CPHA	Signál, jehož nastavením se signálem CPOL se určuje nastavení parametrů mezi hodinovým a datovým signálem
CPOL	Signál, jehož nastavením se signálem CPHA určuje nastavení parametrů mezi hodinovým a datovým signálem
CPU	Označení pro mikroprocesor
CS	Slouží pro výběr obvodu
č/č	Čítač/časovač
DB0 až DB7	Vstupy LCD displeje, na které jsou data zasílána
DD/MM/RR	Zkratka časového formátu den/měsíc/rok
DDRAM	Paměť displeje určená pro uložení pozice znaků
DIL-40	Pouzdro součástky se čtyřiceti vývody
DPH	Horní osmi bitová polovina registru DPTR
DPL	Dolní osmi bitová polovina registru DPTR
DPTR	Registr, který se skládá ze dvou bajtů DPH a DPL. Využívá se jako ukazatel do paměti ROM
DTE	Nepřekřížené sériové vedení ze signálu RxD/TxD
E	Vstup, na kterém se potvrzují platná data u LCD

EA	Signál řešící překrývání vnější a vnitřní paměti
ED2	Vývojová deska sloužící jako řídicí prvek sestavených desek
EEPROM	Elektricky mazatelná paměť typu ROM-RAM. Má omezenější počet zápisů, maže se elektricky
EOSC	Bit umístěný v kontrolním registru aktivující a deaktivující krystal připojený k RTC
EPROM	Typ paměti typu ROM-RAM, jejíž obsah je mazatelný ultrafialovým zářením(UV).
Expandér	obvod umožňující rozšíření vstupů a výstupů na procesoru
FET	Polem řízené unipolární tranzistory
FLASH	Je to typ paměti typu RAM s náhodným přístupem
Full-duplex	Plně duplexní
UART	Univerzální asynchronní vysílač a přijímač
GND	Zem
HH/MM/SS	Zkratka časového formátu hodina/minuta/sekunda
HIGH	Logická úroveň 1
IEN0	Registr povolení nebo zakázání jednotlivých přerušení
INTCN	Bit umístěný v kontrolním registru umožňující povolení přerušení dvou alarmy u RTC
INTn	Vstup vnějšího přerušení
IPL0 a IPH0	Registry sloužící k nastavování priority přerušení
IRQF0	Bit umístěný ve stavovém registru umožňující dotazování alrmu0 u RTC
IRQF1	Bit umístěný ve stavovém registru umožňující dotazování alrmu1 u RTC
JUMPER	Propojka sloužící k propojení dvou pinů na svorkovnici
Katoda	Představuje elektrodu se záporným napětím
KO D	Klopný obvod typu D se dvěma stabilními stavy
KO	Klopný obvod
LCD	Zkratka označující LCD-displej
LOW	Logická úroveň 0
MASTER	Obvod označován jako řídicí
MISO	Master čte data od všech zařízení, která jsou slave u sběrnice SPI
MOSI	Master posílá data do všech zařízení, která jsou slave u sběrnice SPI
NPN	Jedná se bipolární tranzistor s přechodem NPN
OSC	Oscilátor slouží k nastavení taktovací frekvence

P1	Port procesoru 1 sloužící ke komunikaci procesoru s připojeným zařízením
P1.4	Jedná se o port P1 a čtvrtý bit
P2	Port procesoru 2 sloužící ke komunikaci procesoru s připojeným zařízením
P3	Port procesoru 3 sloužící ke komunikaci procesoru s připojeným zařízením
P4	Port procesoru 4 sloužící ke komunikaci procesoru s připojeným zařízením
PAL	Programovatelný obvod
GAL	Programovatelný obvod
FPGA-	Programovatelný obvod
PC	Čítač instrukcí
PCA	Programovatelné čítačové pole.
PNP	Označení přechodů tranzistoru
PSEN	Řídící signál mikroprocesoru používaný při čtení s vnější pamětí
PWM	Pulzně šířková modulace jedná se o druh řízení motorů pomocí změn šířky pulzů
R/W	Vstup u LCD-displeje určují zdali se jedná o čtení či zápis dat
R0 až R7	Pomocné registry sloužící k manipulování s daty a adresami
RAM	Paměť s náhodným přístupem používaná jako paměť dat
RD	Signál z procesoru sloužící pro čtení dat.
RET	Návrat z podprogramu
RI	Signál určující přerušení od sériové linky příjem
ROM	Jedná se o paměť do, které jsou data uložena při výrobním procesu
RS	Vstup u LCD-displeje určují zdali se jedná o instrukci či data
RS232	Sériové rozhraní sloužící ke komunikaci mezi ED2 a počítačem
RST	Signál z procesoru sloužící pro vynulování procesoru
RTC	Obvod reálné času
SCK	Hodinový signál
SCLK	Hodinový signál
SDI	Bit pro vstupní sériová data například u RTC
SDO	Bit pro výstupní sériová data například u RTC
SERMODE	Bit určující typ výběru komunikace u RTC

SFR	Speciální funkční registr procesoru
SLAVE	Obvod označován jako podřízený
SMD	Označení pro miniaturní součástky
SPI	Sériové vnější rozhraní slouží pro komunikaci např. procesoru a obvodu RTC
TCON	Registr umístěný v SFR, sloužící k řízení čítače/časovače
TMOD	Registr umístěný v SFR, sloužící ke konfiguraci čítače/časovače
TTL	Jedná se o typ obvodů, který využívá tranzistorovou logiku
TUL	Technická univerzita v Liberci
v/v	Vstupní/výstupní porty mikroprocesoru
VDT1	Výuková deska 1
VDT2	Výuková deska 2
VDT3	Výuková deska 3
watchdog	Má za úkol přivést systém ze zaseknutého stavu k normální funkci např. vyresetováním procesoru
WP	Bit umístěný v kontrolním registru umožňující zapsání dat do paměti RAM u RTC
WR	Signál z procesoru sloužící pro zápis dat
XRAM	Vnější paměť typu RAM
XTAL1	Pin na procesoru k němuž se připojuje taktovací krystal
XTAL2	Pin na procesoru k němuž se připojuje taktovací krystal

Úvod

Úkolem této bakalářské práce bylo k dané vývojové desce ED2 vytvořit tři samostatné moduly, které budou sloužit studentům Technické univerzity v Liberci (dále jen „TUL“) při výuce mikroprocesorové techniky. Při praktické výuce na vytvořených přípravcích získají studenti potřebnou praxi a znalosti v oblasti programovatelných jednočipových mikroprocesorů řady 8051 a jejich klonů. Hlavním cílem bylo vytvořit výukové desky co nejjednodušší, a to jak z hlediska pochopení návrhů připojování periférií, tak i po stránce programování výukových desek. Díky těmto výukovým deskám mohou studenti bez obav realizovat své myšlenky při programování v reálném čase na skutečném přípravku, aniž by způsobili škody v pracovním procesu.

V první kapitole je popsána činnost a funkce vývojové desky ED2, na které je umístěn mikroprocesor od firmy ATMEL AT89C51ED2, který tvoří srdce celého projektu.

Druhá kapitola seznámí čtenáře s obecnými vlastnostmi a funkcemi mikroprocesorů řady 8051 a vybraného mikroprocesoru AT89C51ED2.

Třetí kapitola pojednává o první vytvořené desce (VDT1), která byla konstruována pro začátky programování jednočipových mikroprocesorů při výuce na TUL. A to na přípravcích LCD, klávesnici a L-Bargrafu.

Čtvrtá kapitola se zabývá konstrukcí druhé výukové desky (VDT2), která byla sestrojena za účelem seznámení studentů se segmentovými displeji, tlačítky, reproduktorem a obvodem reálného času (RTC), který komunikuje s procesorem po SPI rozhraní.

Pátá kapitola je věnována konstrukci poslední výukové desky (VDT3). Díky této desce se studenti seznámí s jednoduchým programováním (řízením) krokových motorů a vyzkouší si jeho funkci společně se segmentovým zobrazovačem a tlačítky.

Poslední šestá kapitola nabízí ukázky jednotlivých funkčních vzorových úloh s fragmenty zdrojového kódu.

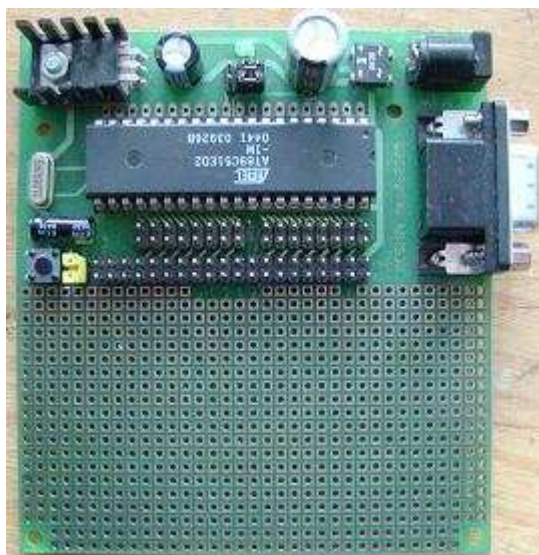
1. Vývojová deska ED2

Vývojová deska ED2, ke které jsou připojeny výukové desky, byla vyrobena firmou Promis. K této desce lze pomocí konektorů připojit jakékoliv zařízení, které splňuje podmínky pro správnou funkci celého kompletu. Mezi důležité vlastnosti, které je třeba dodržet při navrhování a připojování zařízení k desce ED2, patří např.: proudové zatížení pinů procesoru, stabilizátoru napětí a správné zapojení portů mikroprocesoru.

Bylo řečeno [1], že se jedná o univerzální prototypovou vývojovou desku pro procesory řady Atmel AT89C51xD2 (Temic, Philips). Obsahuje pole, které je možné osadit několika jednoduchými obvody. Pro připojení k hostitelskému počítači lze použít tří vodičový kabel RS-232 DTE (nepřekřížené vedení RxD/TxD) nebo převodníky USB/RS-232, pokud počítač nemá interface RS-232 (např. notebooky). Ke komunikaci mezi deskou a počítačem je možné použít plnou verzi programu uScope nebo FLIP.

Na desce jsou plošky pro osazení konektoru, umožňujícího zasunutí destičky do patice procesoru jiné aplikace. Tímto způsobem je vývojovou desku ED2 možno využít jako levný ladící prostředek pro aplikace osazené jiným typem procesoru, případně pro aplikace, kde není vyveden sériový port. Výuková deska ED2 je napájena pomocí síťového adaptéru 230/7-9V s vyvedeným kladným pólem na prostřední kolík. Přítomnost napájecího napětí signalizuje dioda.

Na desce je v patici umístěn procesor AT89C51ED2 v pouzdře DIL-40. Díky snadnému vyndávání procesoru se tato deska stává rovněž jednoduchým a levným programátorem pro procesory AT89C51xD2. K procesoru je připojen krystal taktovacího kmitočtu o frekvenci 11,0592 MHz. K procesoru je rovněž připojeno tlačítko, díky kterému je možné provést vynulování (reset) mikroprocesoru. Brány (porty) mikroprocesoru jsou vyvedeny na pinové konektory. Na konektory jsou vyvedeny rovněž důležité signály pro ovládání obvodů, které mohou rozšířit funkci vývojové desky (ALE, PSEN, EA, +5V, GND). Pro možnosti komunikace vývojové desky s počítačem je nutné se zmínit o svorkovnici X15, která je umístěna u resetovacího tlačítka. Na jednom pinu je umístěn signál PSEN a na druhém signál GND. Touto propojkou je možné nastavit komunikační vlastnosti mezi deskou a jednotlivými programovacími programy.



Obr.1: Skutečné zobrazení desky ED2

Procesor AT89C51ED2 od firmy Atmel [2], který má na starosti řízení připojených obvodů, je plně kompatibilní s procesory x51. Tento procesor se vyznačuje oproti klasickým procesorům typu x51 novými a vylepšenými vlastnostmi.

1.1 Základní vlastnosti a parametry procesoru AT89C51ED2

- Vývojově i instrukčně kompatibilní s 8051
- 256 B interní RAM
- 64 kB programové paměti FLASH
- 1792 B datové paměti XRAM (přístupná přes instrukci MOVX)
- 2 kB paměti EEPROM (může posloužit k uložení nastavení)
- Dva registry DPTR
- X2 - jádro pracuje s rychlostí 6 period hodin na strojový cyklus (namísto standardních 12), pro zachování kompatibility je možné zapnout děličku 2 mezi vstupem XTAL1 a vlastním jádrem procesoru (z hlediska programu se pak procesor chová, jako by měl 12 period hodin na strojový cyklus). Po resetu je dělička zapojena.
- PCA (programovatelné čítačové pole) – umožňuje generování PWM, jednotky capture/compare
- Možnost generování signálu ALE pouze během instrukcí MOVX a MOVC (pro snížení rušení)

- Rozhraní SPI
- Rozhraní pro připojení klávesnice k bráně P1
- Tři 16-bitové čítače/časovače
- Devět zdrojů přerušení se čtyřmi úrovněmi priorit
- Integrovaná správa napájení, možnost volby režimu se sníženou spotřebou
- Hardwarový „watch dog“ časovač
- Full-duplex UART s vylepšenými funkcemi

1.2 Komunikace desky ED2 s programovým vybavením

1.2.1 uScope

Aby bylo možné komunikovat s deskou ED2 pomocí softwaru uScope firmy Promis, je nutné provést následující kroky. Spuštění softwaru uScope. Připojit desku ED2 k napájecímu zdroji a sériovým kabelem nebo převodníkem připojit desku k počítači, na němž je nainstalován daný software. Propojka na svorce X15 nesmí být připojena (signál PSEN nesmí být uzemněn). Stisknutím tlačítka RESET na desce ED2 dojde k autodetekci baudové rychlosti. V programu uScope je nutné nastavit v dialogu nastavení komunikace parametry linky RS-232 nebo převodníku. A potvrdit nastavení tlačítkem „*Init connection*“. Návod na cvičení k ovládání programu viz příloha číslo 5.

1.2.2 FLIP

Pro připojení nahrávacího programu FLIP je nutné nejdříve na desce ED2 svorku X15 spojit propojkou (uzemnit signál PSEN). Přivést napájecí napětí na desku ED2 a přes komunikační rozhraní připojit kabel RS-232 nebo převodník USB/RS-232 k desce a počítači. Po spuštění utility FLIP (verze 3.1.1) bude v záložce „*Settings*“ vybrána volba „*Preferences*“. Po otevření dialogového okna bude zvolena záložka „*RS232*“. Po označení bodu s názvem „*ISP Hardware Conditions Controlled by Flip*“ musí být signály zvoleny do těchto úrovní: „*RST(DTR)*“ aktivní pozice „*High*“ a signál „*PSEN(RST)*“ do aktivní pozice „*Low*“. Následně je nutné v ikonovém menu zvolit tlačítko zobrazující čip „*Select a Target device*“ a vybrat typ procesoru AT89C51RD2. Dále bude zvoleno v ikonovém menu tlačítko symbolizující připojení „*Select a Communication medium*“ a zvolena volba RS-232. Po otevření záložky je zvolen komunikační port a rychlost připojení. Důležité je v tento moment vyresetování

mikroprocesoru tlačítkem „Reset“ na vývojové desce ED2. Nyní bude provedeno spojení tlačítkem „Connect“. Po propojení desky ED2 a počítače je v záložce „File“ vybrána volba „Load Hex File“. Je zvolen program, který bude do desky ED2 nahrán. Musí se jednat o již přeložený program (*.hex). Po výběru programu bude spuštěno tlačítkem „Run“ vymazání a nahrání programu do procesoru AT89C51ED2. Program nahraný v desce ED2 bude spuštěn po potvrzení tlačítka „Start application“. Návod na cvičení k programu viz příloha číslo 5.

1.3 Shrnutí technických parametrů desky ED2

- Mikroprocesor AT89C51ED2 v pouzdře DIL-40, Krystal 11,0592 MHz
- RS232 rozhraní s převodníkem úrovní TTL/RS232
- Jumper pro aktivaci Bootloaderu, podpora nahrávání aplikací z počítače do paměti FLASH po RS232
- RESET tlačítko
- Stabilizátor napájecího napětí (7-12 V)
- 2x16-pinový konektor P1 a P3 (každý vývod 2x)
- 38-pinový konektor P2, P0, signály PSEN a ALE
- Rozměry plošného spoje 100 x 75 mm.

Elektrotechnické schéma desky ED2 je samostatnou přílohou (příloha č. 1) tohoto dokumentu.

1.4 Možnosti rozšíření Desky ED2

Bylo řečeno [3], že pokud aplikace procesoru vyžaduje větší paměť, větší počet vstupů/výstupů, případně další pomocné obvody jako jsou hlídací časovač, obvod reálného času, hlídač napájecího napětí apod., je možné k procesoru připojit externí obvody a komunikovat s nimi pomocí bran procesoru například (expandér). Nebo použít externí datovou a adresovou sběrnici procesoru. Ve druhém případě je třeba k zachycení nižšího bytu adresy použít vnější záchytný adresový registr. Některé rozšiřující obvody ve své struktuře obsahují tento registr a jsou pro komunikaci s procesorem přizpůsobeny. Pokud je nutné použít při rozšíření systému další logiku,

např. adresové dekodéry, přepínače, jednoduché obvodové řadiče a podobně, je vhodné tyto obvody implementovat pomocí programovatelných obvodů PAL, GAL, FPGA.

1.4.1 Expandér - specializované obvody (8255,8155)

Při použití expanderu je nutná náročnější softwarová obsluha z důvodu adresování požadovaného vstupního/výstupního režimu expandéru. Vstupy a výstupy obvykle nelze adresovat jednotlivě (po bitech), ale po skupinách, expandéry však poskytují režim logického součinu, čímž lze zajistit adresování jen určitých vstupních/výstupních (bitů).

Pro komunikaci mezi deskou ED2 a externí pamětí programu lze využít řídicí signál mikroprocesoru PSEN pro čtení. Při komunikaci s externí pamětí dat se využívá dvou řídicích signálů - RD (čtení) a WR (zápis). Pro přístup do paměti programu se vždy používá 16-bitová adresa. Pro přístup do externí paměti dat se může používat buď 16-bitová adresa (MOVX @DPTR,A) nebo 8-bitová adresa (MOVX A,@Rr).

Pokud se používá 16-bitové adresace (pro paměť dat), vyšší byte adresy se vysílá na port P2, kde zůstává po celou dobu cyklu čtení nebo zápisu. Výstupní budiče portu P2 v tomto případě využívají zvyšovací odpory po celou dobu, kdy se portem vysílají informace v logické úrovni „1“, které jsou dány adresou. Tak je tomu tedy během vykonávání instrukce MOVX @DPTR,A. Během této doby klopný obvod (KO) portu P2 (v SFR) nemusí obsahovat logické úrovně ve stavu „1“ a obsah KO není ani nijak modifikován. Pokud po cyklu externí paměti okamžitě nenásleduje změna informace v dalším cyklu externí paměti, bude odvyšlán nezměněný obsah KO portu P2. Pokud se používá 8-bitové adresace (MOVX A,@Rr), obsah KO portu P2 (tedy příslušné SFR) zůstane na pinech portu P2 po celou dobu cyklu externí paměti. To usnadňuje stránkování. V jiném případě je na portu P0 časově přepínán (multiplexován) nižší byte adresy s datovým bytem. Vnitřní signál CPU Adresa/Data ovládá oba FET tranzistory ve výstupních budičích portu P0 a pak tedy piny portu nejsou obvodem s otevřeným kolektorem a nevyžadují tím pádem externí zvyšovací rezistory. Signál ALE by měl být použit k zachycení adresového byte do externího KO. Adresový byte je platný při sestupné hraně signálu ALE. Poté se, v cyklu zápisu, datový byte, který má být zapsán, vyšle na port P0 přesně před aktivací signálu WR a zůstane zde, dokud není signál WR deaktivován. Během každého přístupu k externí paměti procesor zapíše FFH do KO portu P0, potom tedy mazání jakékoliv informace v KO portu P0 může být

pozdrženo. Pokud uživatel zapisuje do portu P0 během vybavování dat v externí paměti, příchozí datový byte bude znehodnocen. Proto se nezapisují data do portu P0, pokud se používá externí paměť programu. Přístup k externí paměti programu je umožněn, pokud jsou splněny některé z následujících podmínek:

- 1) signál EA je aktivní
- 2) čítač instrukcí PC obsahuje číslo větší než FFFH

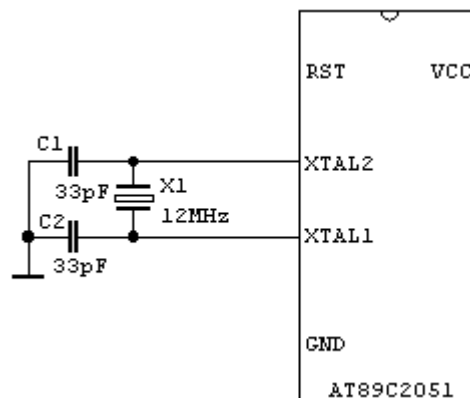
Když mikroprocesor vykonává program v externí paměti, všechny piny portu P2 jsou nastaveny jako výstupní a nemohou být používány pro normální použití vstupů/výstupů. Během vybavování dat externího programu port P2 vysílá vyšší byte čítače instrukce (PC).

2. Vlastnosti mikroprocesoru AT89C51ED2 a řady 8051

2.1 Časování a nulování mikroprocesoru

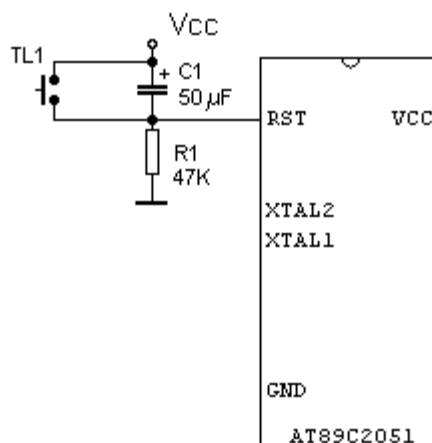
Bylo řečeno [4], že mikroprocesor ke své činnosti a správné funkci potřebuje zdroj hodinových „taktovacích“ impulsů o určité frekvenci (f_{osc}), které využívají vnitřní obvody mikroprocesoru a od kterého je odvozena délka trvání strojového cyklu a tedy i doba vykonávání instrukcí. K taktování mikroprocesoru může být využito buď vnitřního obvodu oscilátoru, který je zabudován přímo v čipu mikroprocesoru, nebo může být použit externí obvod oscilátoru. Na desce ED2 byla použita varianta s vnitřním oscilátorem s připojením krystalu. Pokud tedy k taktování mikroprocesoru bude použita varianta s vnitřním oscilátorem s připojením krystalu, platí: frekvence použitého krystalu(f_{psc}) = frekvence oscilátoru (f_{osc}).

Zdrojem taktovacích impulsů může být piezokeramická jednotka připojená k mikroprocesoru nebo jiný externí zdroj taktovacích impulsů.



Obr.2: Taktování vnitřním oscilátorem – připojení krystalu

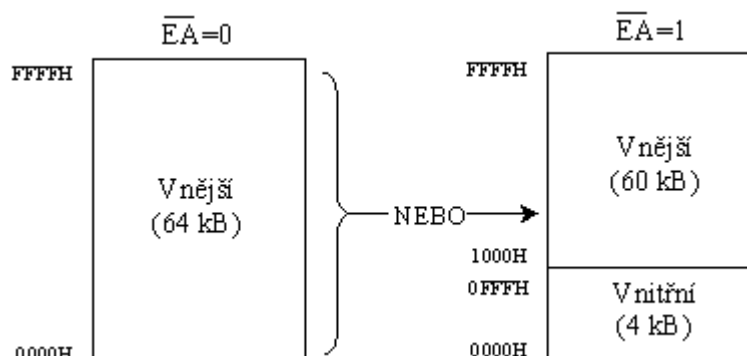
Pro inicializaci mikrokontroléru je k dispozici vstup RST (reset), na který je připojen interní Schmittův obvod. Proto lze na tento vstup připojit i spojitě se měnící napětí, např. napětí na kondenzátoru. Při automatické inicializaci po připojení napájecího napětí. Pro verze CMOS lze rezistor vynechat, protože vstup RST má vnitřní zatěžovací rezistor. Tehdy musí být hodnota předřazené kapacity snížena na úroveň asi 6μF. Procesor AT89C51ED2 [2] má reset mikroprocesoru v logické úrovni 1 (obr. 3). Je třeba upozornit také na to, že řada nástupců 8051 má aktivní úroveň nulovacího impulsu opačnou (log 0).



Obr.3: Resetovací obvod v aktivní úrovni. $RST=log1$

2.2 Organizace paměti

Bylo řečeno [5], že mikroprocesory řady 8051 mají oddělené paměťové prostory - paměť programu (ROM, příp. EPROM, EEPROM, FLASH) a paměť dat (RAM). Tyto paměťové prostory jsou přístupné různými instrukcemi. Paměťový prostor může být dále vnitřní, umístěný přímo na čipu mikroprocesoru, nebo vnější, umístěn mimo čip a realizován prostřednictvím dalších externích obvodů. U základního mikroprocesoru 8051 má vnitřní paměť programu velikost 4 kB a vnější paměť má maximální velikost 64 kB. Mikroprocesor je vybaven vstupem EA, který řeší překrývání vnitřního a vnějšího paměťového prostoru (viz obr. 4). Je-li vstup EA=0, potom paměť programu je tvořena celou vnější pamětí. Je-li vstup EA=1, potom se instrukce v paměťovém prostoru 0000H až 0FFFH čtou z vnitřní paměti programu a mimo tento prostor (tedy z adresy 1000H až FFFFH) ze zbývajících 60 kB vnější paměti programu.



Obr.4: Struktura paměťového prostoru programu

Procesor AT89C51ED2 [2] má oproti obyčejné 8051 následující paměťové parametry:

- 256 B interní RAM
- 64 kB programové paměti FLASH
- 1792 B datové paměti XRAM (přístupná přes instrukci MOVX)
- 2 kB paměti EEPROM (může posloužit k uložení nastavení)

2.2.1 Přímé a nepřímé adresování

Přímé adresování - adresa přímo udává paměťové místo, se kterým se bude dále pracovat. Například MOV A,20H do akumulátoru uloží obsah bytu na adrese 20H.

Nepřímé adresování - danou instrukcí není přímo určena adresa paměťového místa (bytu nebo bitu), ale je odkazováno na registr, ve kterém je adresa uložena. Při zápisu se před tento registr píše „@“. Například MOV A,@R1, kde je v R1 uložená konstanta udávající adresu, z které se bude číst byte a ten se následně uloží do akumulátoru.

2.3 Čítače a časovače

Bylo řečeno [6], že čítače a časovače tvoří nepostradatelnou součást mikroprocesoru. Čítač spravuje určité paměťové místo, ke kterému přičítá jedničku na základě zjištění náběžné nebo sestupné hrany sledovaného signálu. Sledovaný signál je u čítače určitý vnější signál, například výstupní signál z indukčního nebo fotoelektrického čidla (snímače).

Časovač spravuje určité paměťové místo, ke kterému přičítá jedničku, funguje také jako čítač, ale na rozdíl od čítače je zde sledovaným signálem určitý vnitřní signál se známým průběhem o konstantní frekvenci (tzv. hodinový signál). Jeho použití je nejčastěji ve funkci vytvoření určitého časového intervalu, například zpoždění.

Mikroprocesor AT89C51ED2 [2] má tři 16-bitové čítače/časovače na rozdíl od standardní 8051, která má pouze dva 16-bitové čítače/časovače. Konfigurace čítače/časovače je prováděna nastavením registru TMOD, řízení čítače/časovače je prováděna nastavením registru TCON, oba tyto registry se nacházejí v SFR (speciální funkční registr). U procesoru AT89C51ED2 je třetí čítač/časovač nastavován a řízen registry T2MOD a T2CON umístěnými v SFR. Tento čítač/časovač může pomocí nastavení v registru T2MOD čítat vzestupně i sestupně. Jeho přetečení nastane v FFFFh

a nastaví bit pro přerušení. U tohoto čítače/časovače je rovněž možno zakázat nastavování přerušení.

2.4 Přerušovací systém

Bylo řečeno [7], že přerušovací systém umožňuje komunikaci mezi probíhajícím hlavním programem a určitými hardwarovými částmi mikroprocesoru. Při běhu hlavního programu v určitém okamžiku (v závislosti na typu zdroje přerušení) vyšle daná hardwarová část procesoru požadavek do CPU (přes řadič přerušení), že chce být obsloužena. Je-li přerušení od dané hardwarové části povoleno, procesor přeruší vykonávání hlavního programu a začne vykonávat program zajišťující obsluhu vzniklého přerušení. Je-li program obsluhy přerušení dokončen, procesor se vrací k vykonávání hlavního programu od místa, ve kterém skončil před přechodem do programu obsluhy přerušení. Pokud však dané přerušení není povoleno, procesor ho neobslouží a pokračuje dále ve své předchozí činnosti.

Procesor AT89C51ED2 [2] má k dispozici celkem 9 možných druhů přerušení ve čtyřech možných úrovních. Na rozdíl od obecné 8051, kde je pouze 5 možných druhů přerušení ve dvou možných úrovních.

Jedná se o tyto druhy přerušení: dvě externí přerušení (INT0, INT1) přerušení od tří čítačů časovačů (č/č0, č/č1 a č/č2), přerušení od sériového portu, přerušení od sběrnice SPI, přerušení od klávesnice a globální přerušení.

Vznik přerušení je během vykonávání hlavního programu neustále procesorem monitorován (zjišťuje stavy příznakových bitů, které jsou k tomuto účelu u daných hardwarových částí určeny).

V obsluze přerušení se mohou používat cykly, skoky, volání atd. stejně jako v hlavním programu. Nedoporučuje se volat (pomocí CALL a RET) z obsluhy přerušení podprogram umístěný v hlavním programu z důvodu hrozícího konfliktu.

Procesor umožňuje povolit nebo zakázat vybrané zdroje přerušení. K tomuto účelu je u mikroprocesoru uživateli k dispozici registr povolení přerušení IEN0 Protože mohou přijít žádosti o přerušení i od devíti zdrojů naráz, musí být zajištěno postupné zpracování všech vzniklých přerušení. Toto zabezpečuje systém priority, který jasně řekne, které přerušení bude obslouženo jako první, které jako druhé a které bude obslouženo jako poslední. U mikroprocesoru je systém priority přerušení řešen tak, že

jednotlivé zdroje přerušení mají pevně přidělenou prioritu v každé ze čtyř programově volitelných úrovní priorit (viz obr 5).

Číslo	Priorita	Zdroj přerušení	Dotaz na přerušení	Adresa
0	0	Reset		0000h
1	1	INT0	IE0	0003h
2	2	Timer 0	TF0	000Bh
3	3	INT1	IE1	0013h
4	4	Timer 1	IF1	001Bh
5	5	UART	RI+TI	0023h
6	6	Timer 2	TF2+EXF2	002Bh
7	7	PCA	CF+CCfn (n=0-4)	0033h
8	8	Keyboard	KBDIT	003Bh
9	9	-	-	0043h
10	10	SPI	SPIIT	004Bh

Obr.5: Přirozená priorita u AT89C51ED2

Díky pevnému stanovení priorit známe přesný sled zpracování vzniklých přerušení. Někdy pevně daný sled nemusí vyhovovat požadavkům pro řízení aplikace. A proto lze jednotlivým zdrojům přerušení nastavit odpovídající prioritu v registrech speciálních funkcí IPL0 a IPH0.

2.5 Porty

Bylo řečeno [8], že vstupně/výstupní porty umožňují připojit k mikroprocesoru zařízení, například (LED-diody, LCD displej či tlačítka). Vstupně/výstupní (dále jen „v/v“) porty jsou slučitelné s TTL/CMOS logikou, což udává napěťové úrovně pro log.0 a 1.

U standardní verze 8051 jsou k dispozici 4 v/v porty, z nichž každý má po 8 pinech a u procesoru AT89C51ED2 jsou k dispozici u pouzdra DIL-40 také 4 v/v porty. U procesoru s pinovým pouzdem 64 a 68 je však těchto vstupních a výstupních portů 6. Namísto slova „port“ se někdy lze setkat s označením *brána*. Slovem *port/brána* je myšlen celý port, tedy všech 8 pinů jemu příslušejících. Tyto čtyři porty jsou označovány P0, P1, P2 a P3 (P4,P5). Označení konkrétního pinu určitého portu je následující – např. P1.4, kde 1 před desetinnou tečkou určuje port a 4 za tečkou je číslo konkrétního pinu příslušejícího k portu P1. Jelikož se jednotlivé piny adresují jako bity, je možné setkat se s označením *bit*, např. bit P1.5.

Všechny v/v porty jsou obousměrné. Jednotlivé piny všech portů v rámci jednoho programu lze používat jako vstupní i jako výstupní.

Port P0 jako jediný nemá ve výstupním budiči zvyšovací (zdvihací) odpory (pull up)- jde tedy o obvod s otevřeným kolektorem. Pokud bude použit jako obecný v/v port, musí být k pinům připojeny externí odpory, jejichž druhý konec bude připojen na +Ucc. Port P0 může být využit také při styku s vnější pamětí.

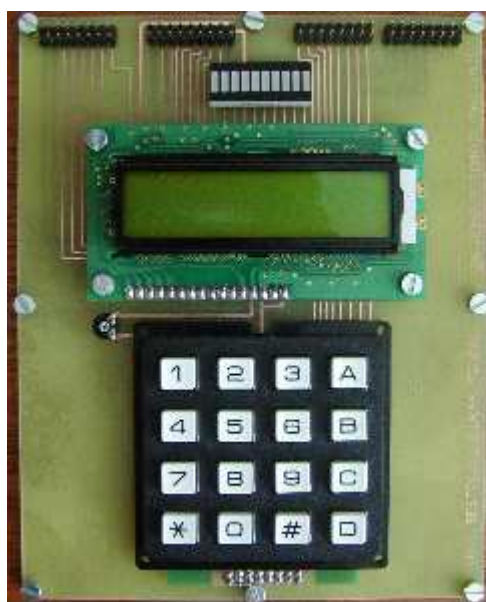
Port P1 může plnit speciální funkce (např. připojení klávesnice apod.). Tato funkce je dána výrobou dle jednotlivých typů procesorů. Způsob využití portu P1 není nijak omezen.

Port P2 kromě své normální funkce může být využit společně s portem P0 ke komunikaci s externí pamětí. K tomuto účelu se využívá jen části obou portů (výstupní budiče portu P0 a P2 a vstupní vyrovnávací paměť portu P0). Jinak je možno na port P2 zapojit jakékoli zařízení bez omezení.

Port P3 - všechny piny portu P3 mohou mít více funkcí. Kromě standardní funkce navíc tento port poskytuje své piny pro potřeby jiným vnitřním obvodům mikroprocesoru. Alternativní funkci pinů portu P3 lze aktivovat pouze v případě, kdy příslušný bit v SFR obsahuje log.1. Jinak pin zůstává v log.0.

3. První výuková deska VDT1

Tato deska byla navržena a postavena za účelem zkvalitnění výuky mikroprocesorové techniky na Technické univerzitě v Liberci. Hlavním cílem bylo přiblížit studentům připojování a programování základních prvků (součástek) používaných v mikroprocesorové technice. Pro snadnější pochopení učiva bylo vytvořeno několik vzorových příkladů včetně jejich řešení. Na obrázku 6 jsou umístěny prvky, které se studenti naučí programovat při výuce. Jsou to: klávesnice, L-Bargraf a LCD displej. Celkové elektrotechnické schéma zapojení desky VDT1 je umístěno v příloze č. 2 tohoto dokumentu.

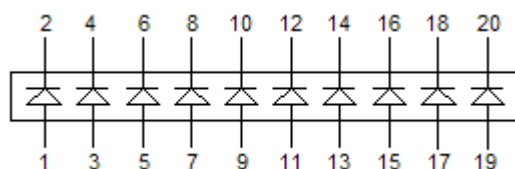


Obr.6: Deska VDT1

3.1 Připojování součástek na desce VDT1

3.1.1 L-Bargraf

Součástka L-Bargraf se skládá z určitého počtu diod umístěných v jednom pouzdře (obr. 7). L-Bargrafů se průmyslově vyrábí velké množství v různých provedeních, s různým počtem jednotlivých diod a různých barev. Taktéž možností využití v průmyslových aplikacích je mnoho. Například při signalizaci stavu paliva v nádrži, signalizaci připravenosti přístroje apod.



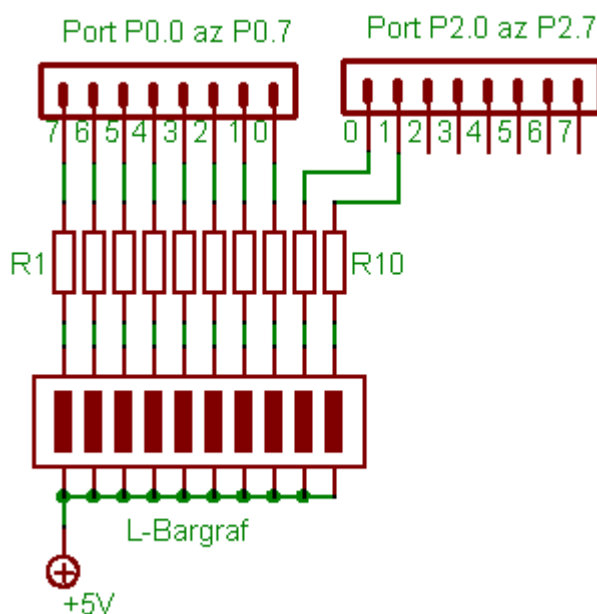
Obr.7: Schématické znázornění zapojení L-Bargrafu

Jak je patrné z obr. 7, každá dioda má svou anodu a katodu. A tak se schématické zobrazení neliší od skutečného zobrazení této součástky.

Pro výukovou desku VDT1 byl zvolen L-Bargraf zelené barvy s počtem deseti diod. Důvodem byla možnost využitelnosti volných pinů procesoru. Dalším z důvodů, proč byl vybrán právě deseti diodový L-Bargraf, bylo záměrné připojení dvou diod na jiný port, aby si studenti vyzkoušeli práci se součástkou, jež je ovládána přes dva porty.

3.1.1.1 Připojování L-Bargrafu

L-Bargraf je zapojen tak, že všechny anody jsou spolu propojeny na napájecí napětí +5V a jednotlivé katody jsou přes odpory připojeny na konektor (obr.8). Konektor je s piny mikroprocesoru propojen přes bránu portu P0.0 až P0.7 a P2.0 až P2.1. Odpory jsou zvoleny s ohledem na proudové zatížení jednotlivých diod umístěných v součástce L-Bargraf.



Obr.8: Schématické zapojení součástky L-Bargraf

Výpočet:

Každá samostatná led dioda v L-Bargrafu svítí při hodnotě $U_d = 1,6 \text{ V}$ (uvádí výrobce součástky). Proud, který prochází každou diodou součástky L-Bargraf, je $I_d = 10,5 \text{ mA}$. Hodnota napájecího napětí je $U_{cc} = +5\text{V}$. Následuje výpočet odporů R_1 až R_{10} pomocí Ohmova zákona a jejich jmenovitého zatížení P (obr.8).

$U_{cc} = 5 \text{ V}$; $U_D = 1,6 \text{ V}$; $10,5 \text{ mA}$; $R_b = ?$; $P = ?$

$$R_b = (U_{cc} - U_d) / I_d = (5 - 1,6) / 0,0105 = \underline{323,8 \Omega} \Rightarrow \underline{330 \Omega}$$

Vypočtená hodnota předřadného odporu je rovna **323,8 Ω** . Z katalogu elektro-technických součástek je vybrána nejbližší vyšší hodnota odporu. V tomto případě je odpor roven **330 Ω** .

$$P = (U_{cc} \cdot I_d) = (5 \cdot 0,0105) = \underline{52,5 \text{ mW}}$$

Z výpočtu je patrné, že lze použít odpory SMD o velikosti 0805, které mají jmenovité zatížení $0,125 \text{ W}$ při 70°C .

3.1.1.2 Programování L-Bargrafu

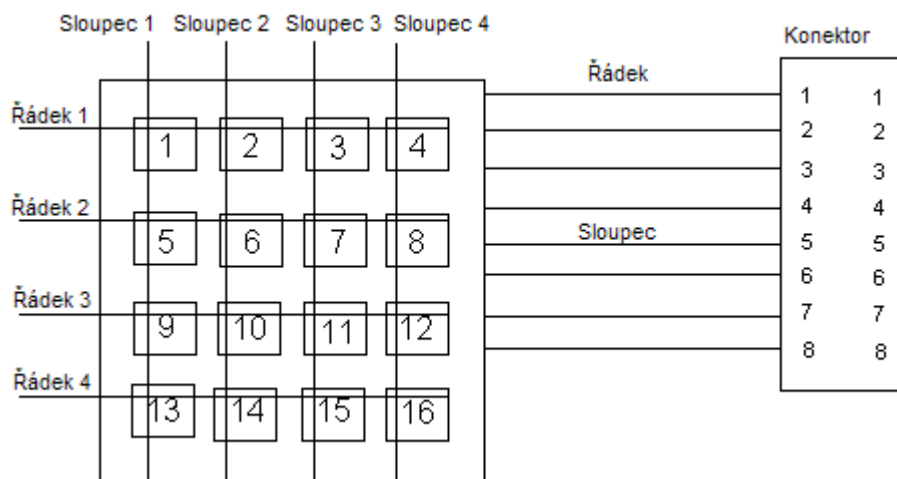
Jednotlivé diody jsou spínány z výstupu mikroprocesoru, a to tak, že z procesoru je vyslána logická úroveň o hodnotě 0 a tím dojde k uzemnění dané katody a dioda se rozsvítí a naopak. Ke zhasnutí musíme na daný pin přivést z mikroprocesoru logickou úroveň 1 tedy $+5\text{V}$.

3.1.2 Maticová klávesnice na desce VDT1

Klávesnice jako ovládací prvek je v průmyslu využívána například v těchto aplikacích: ovládání CNC strojů; nastavování kontrolních hodnot na měřicích pracovištích apod.

3.1.2.1 Připojování klávesnice

Klávesnice k mikroprocesoru je připojena maticovým způsobem. Pokud bude potřeba ovládat (řídit) přípravek šestnácti klávesami, není nutné každou klávesu připojovat na samostatný pin portu. Bylo by tak zbytečně obsazeno velké množství pinů, které lze využít jiným způsobem. Proto se využívá připojování klávesnice maticovým způsobem.



Obr.9: Schéma maticově snímané klávesnice

Z konstrukce zapojení (obr.9) je patrné, že při snímání stavu tlačítek nebude každá klávesa připojena samostatně na pin procesoru. Všechny sloupce i řádky jsou dány čtyřmi bity. Jejich vzájemná kombinace při stisku určité klávesy určí, o jakou klávesu se jedná. Na desce VDT1 je klávesnice připojena na piny procesoru v bráně P3.0 až P3.7.

3.1.2.2 Programování klávesnice

K příkladu je použit obr.9. Pokud budou bity na konektoru dva a osm v logické úrovni nula 0 - je stisknuta klávesa s číslem osm.

V programu je zpravidla sestavena testovací smyčka, díky níž je zjištěno, která klávesa byla stisknuta. Po jejím stisku bude provedena předem naprogramovaná operace. Vznikne-li potřeba ošetřit klávesnici proti zákmitům tlačítek, je nutné použít softwarové řešení. Ukázka funkčního kódu snímání klávesnice viz kapitola 6.

3.1.3 LCD-Displej na desce VDT1

LCD-Displej se využívá v domácnostech i v průmyslu. Příklady využití: CD přehrávače, pračky, videa, domácí kina, zabezpečovací technika aj. Mezi jeho hlavní výhody patří velikost, hmotnost, malý proudový odběr, dlouhá životnost, výborná ostrost obrazu. Bohužel ani tyto zobrazovače se nevyvarovaly ve svých počátcích jistým nevýhodám, a to zejména teplotním závislostem tekutých krystalů, které při nízkých a vysokých teplotách začaly ztrácet své fyzikální vlastnosti a to způsobilo, že se displeje zpomalovaly nebo přestávaly úplně fungovat.

Na desku VDT1 byl použit LCD-displej, který používá řadič HD44780. Tento LCD-displej nabízí možnost definovat až osm vlastních znaků (čeština, grafické symboly). Umožňuje výběr velikosti znaků (1x16 až 4x40). U displejů, které využívají řadič HD44780 nebo jeden z jeho klonů, se zapojení vývodů zavádí jako průmyslový standard. Displej lze nastavit do jednořádkového nebo dvouřádkového módu. Díky těmto vlastnostem má uživatel usnadněnou práci při výměně a programování displeje.

Organizace paměti rozložení znaků neumožňuje i přes automatickou inkrementaci čítače přímý přechod mezi řádky, protože řádky na sebe přímo nenavazují. Díky tomu je nutné posouvání textu po displeji vždy obsluhovat softwarově.

3.1.3.1 Připojování LCD-displeje

Jako zobrazovací zařízení na desce VDT1 byl zvolen LCD-displej MC1602 [9]. Tento displej má možnost volby mezi použitím jednoho nebo dvou řádků (2x16 znaků). Připojení LCD-displeje k desce je řešeno osmivodičovým (tedy jednodušším) zapojením. Toto zapojení bylo zvoleno z důvodu snazší komunikace s procesorem AT89C51ED2. Důvodem byl jednodušší komunikační protokol a kratší čas strávený v obslužné smyčce přerušení. Zapojení LCD-displeje je uvedeno v elektrotechnickém schématu (viz příloha č. 2). Signály RS, R/W, E, DB0 a DB1 jsou připojeny na port P1.0 až P1.4, signály DB2 až DB7 jsou připojeny na port P2.2 až P2.7. Na následujícím obrázku (obr. 10) je vysvětlena funkce jednotlivých bitů.

Číslo vývodu	Označení	V/V	Význam
16	LED-		Napájení podsvícení, katoda
15	LED+		Napájení podsvícení, anoda
14	DB7	V/V	Data, bit 7 (nejvyšší)
13	DB6	V/V	Data, bit 6
12	DB5	V/V	Data, bit 5
11	DB4	V/V	Data, bit 4
10	DB3	V/V	Data, bit 3
9	DB2	V/V	Data, bit 2
8	DB1	V/V	Data, bit 1
7	DB0	V/V	Data, bit 0 (nejnižší)
6	E	Vstup	Platná data
5	R/W	Vstup	0 = zápis dat do LCD 1 = čtení dat z LCD
4	R/S	Vstup	0 = vstup je instrukce 1 = vstup jsou data
3	V0		Nastavení kontrastu
2	GND		0V (napájení)
1	Ucc		+5V (napájení)

Obr.10: Funkce vývodů LCD-displeje

3.1.3.2 Znaková sada LCD-displeje

Znaková sada LCD Displeje

B7.....B0	0	0	0	0	0	0	1	1	1	1	1
xxxx0000			0	@	P	`	P	-	9	3	α
xxxx0001		!	1	A	Q	a	q	.	7	4	ä
xxxx0010		"	2	B	R	b	r	'	ι	7	ß
xxxx0011		#	3	C	S	c	s]	U	7	ε
xxxx0100		\$	4	D	T	d	t	\	ε	7	μ
xxxx0101		%	5	E	U	e	u	.	ο	7	ü
xxxx0110		&	6	F	V	f	v	υ	カ	ニ	ρ
xxxx0111		'	7	G	W	g	w	フ	キ	ラ	q
xxxx1000		(8	H	X	h	x	イ	ク	ネ	リ
xxxx1001)	9	I	Y	i	y	ウ	ケ	ル	リ
xxxx1010		*	:	J	Z	j	z	エ	コ	ハ	レ
xxxx1011		+	;	K	[k	[オ	サ	ヒ	ロ
xxxx1100		,	<	L	¥	l	¥	ハ	シ	フ	ワ
xxxx1101		-	=	M]	m]	ユ	ス	ハ	ン
xxxx1110		.	>	N	^	n	^	ヨ	セ	ホ	°
xxxx1111		/	?	O	_	o	_	ツ	マ	°	ö

Adresování jednotlivých znaků se provádí podobným způsobem, jako u již dříve zmíněné maticové klávesnice. Sloupec je volen prvními čtyřmi bity a příslušný řádek druhými čtyřmi bity. Je-li potřeba zobrazit písmenko „velké A“, je z tabulky (obr.11) předem daná bitová kombinace **01000001**.

3.1.3.3 Programování LCD-displeje

Aby bylo možné jednotlivé znaky zobrazovat na LCD-displeji, musí být programátor seznámen s pozicemi, na které je možné jednotlivé znaky ukládat. To je důležité pro zobrazování srozumitelných textů.

Paměť displeje pro uložení pozice znaků je označována DDRAM (Display Data Random Acces Memory). Dále jsou uvedeny pozice, na které lze ukládat znaky pro zobrazení na řádcích (nižší adresa odpovídá znaku více vlevo).

DDRAM Adresa

Adresa D: 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

Adresa Ř1: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Adresa Ř2: 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F

První řádek označen Adresou D určuje pozici na řádku displeje. Jedná-li se o pozici první či poslední (patnáctou). Druhý řádek určuje adresu, na kterou je poslán kód pro daný zobrazovací znak na prvním řádku. Tato adresa je zobrazena v šestnáctkové soustavě. Třetí řádek určuje to samé jako druhý řádek s tím rozdílem, že poslaný znak se zobrazí na druhém řádku LCD-displeje.

Pro zobrazení písmenka „velké A“ na první pozici LCD-displeji je na adresu **00h** zaslána binární kombinace písmene = **01000001b**.

Pro zobrazení stejného písmenka na poslední pozici druhého řádku je na adresu **4Fh** zaslána binární kombinace písmene **01000001b**.

Jak je patrné ze zobrazení adres DDRAM řadič, který je použit v displeji umožňuje zapsat na řádek šestnáct znaků i přesto, že na displeji je zobrazeno pouze osm znaků v každém řádku.

Snadnější ovládání a konfigurování vlastností LCD-displeje umožňují speciální funkce LCD-displeje s kódovým označením. Kódové označení určuje základní vlastnosti a funkce, jež jsou u LCD-displeje k dispozici (obr.12).

Příkaz	Kód										Vysvětlivky	Délka vykonání instrukce
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Smaže displej	0	0	0	0	0	0	0	0	0	1	Smaže displej a nastaví kurzor na pozici 0.	1.64mS
Nastaví kurzor na začátek	0	0	0	0	0	0	0	0	1	*	Nastaví kurzor na pozici 0 a vynuluje posun displeje (DDRAM beze změny)	1.64mS
Nastaví vstupní režim	0	0	0	0	0	0	0	1	I/D	S	Určí směr pohybu kurzoru (I/D) a posun displeje (S). Tyto operace se provádějí během čtení/zápisu.	40uS

Zapne/vypne displej, kurzor a jeho blikání	0	0	0	0	0	0	1	D	C	B	Zapíná/vypíná displej (D), kurzor (C) a jeho blikání (B).	40uS
Nastaví pohyb kurzoru/displeje	0	0	0	0	0	1	S/C	R/L	*	*	Nastaví pohyb kurzoru nebo displeje (S/C) a směr pohybu (R/L). Obsah DDRAM zůstane beze změny.	40uS
Nastavení interface	0	0	0	0	1	DL	N	F	*	*	Nastaví délku interface (DL), počet řádků displeje (N) a znakový font (F).	40uS
Nastaví pozici v CGRAM	0	0	0	1	Adresa v CGRAM					Po tomto příkazu jsou data ze vstupu zaznamenávána do CGRAM namísto DDRAM.		40uS
Nastaví pozici v DDRAM	0	0	1	Adresa v DDRAM					Po tomto příkazu jsou data ze vstupu zapisována do a čtena z DDRAM.		40uS	
Čte příznak BUSY a hodnotu adresového čítače	0	1	BF	DDRAM address					Čte příznak BUSY (BF) indikující, že displej ještě provádí některou operaci, a pozici ukazatele adresy .		0uS	
Zapíše do DDRAM nebo CGRAM.	1	0	Data					Zapíše data ze vstupu DDRAM nebo do CGRAM.		40uS		
Čte data z DDRAM nebo z CGRAM.	1	1	Data					Čte data z aktuální adresy DDRAM nebo CGRAM.		40uS		

Obr.12: Předdefinované funkce LCD-displeje

V tomto obrázku (obr.12) jsou použita písmena pro možnosti volby uživatele. Následující obrázek (obr.13) uvádí přehled a vysvětlivky jednotlivých písmen z předchozího obrázku.

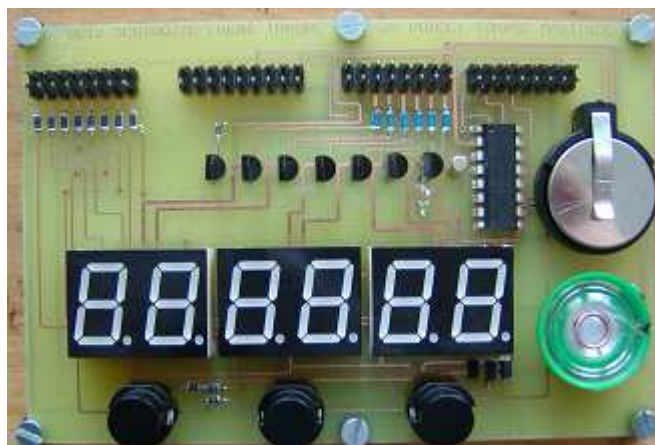
Jméno bitu	Nastavení	
I/D	0 = Snížení pozice kurzoru	1 = Zvýšení pozice kurzoru
S	0 = displ. nepohybovat	1 = displ. Pohybovat
D	0 = Display vypnut	1 = Display zapnut
C	0 = kurzor vypnut	1 = kurzor zapnut
B	0 = blikání kurzoru vypnuto	1 = blikání kurzoru zapnuto
S/C	0 = pohybovat kurzorem	1 = pohybovat displejem
R/L	0 = rotovat vlevo	1 = rotovat vpravo
DL	0 = 4-bit interface	1 = 8-bit interface
N	0 = 1 řádkový displej	1 = 2 řádkový displej
F	0 = 5x7 bodů	1 = 5x10 bodů
BF	0 = může být zaslána instrukce	1 = Displej není připraven, provádí se vnitřní operace.

Obr.13: Význam jednotlivých bitů v bytech instrukcí LCD-displeje

Ukázku zdrojového kódu naleznete v kapitole 6.

4. Druhá výuková deska VDT2

Výuková deska VDT2 byla postavena jako prostředek pro snadné seznámení studentů s programováním a zapojováním segmentových displejů, jednotlivých tlačítek reproduktoru a získávání hodnot reálného času z obvodu RTC po sběrnici SPI. Škála možného využití desky VDT2 je velická, od jednoduchých aplikací až po aplikace náročnější. Elektrotechnické schéma zapojení desky VDT2 je samostatnou přílohou č. 3 tohoto dokumentu.



Obr.14: Deska VDT2

4.1 Seznámení s připojováním součástek k desce VDT2

4.1.1 Segmentové displeje na desce VDT2

Na výukové desce VDT2 jsou umístěny sedmsegmentové displeje o velikosti 14,2 mm typ HD-A542D. Tyto segmentové displeje jsou dvoumístné, to znamená, že v jednom pouzdře se nacházejí dva sedmsegmentové displeje.

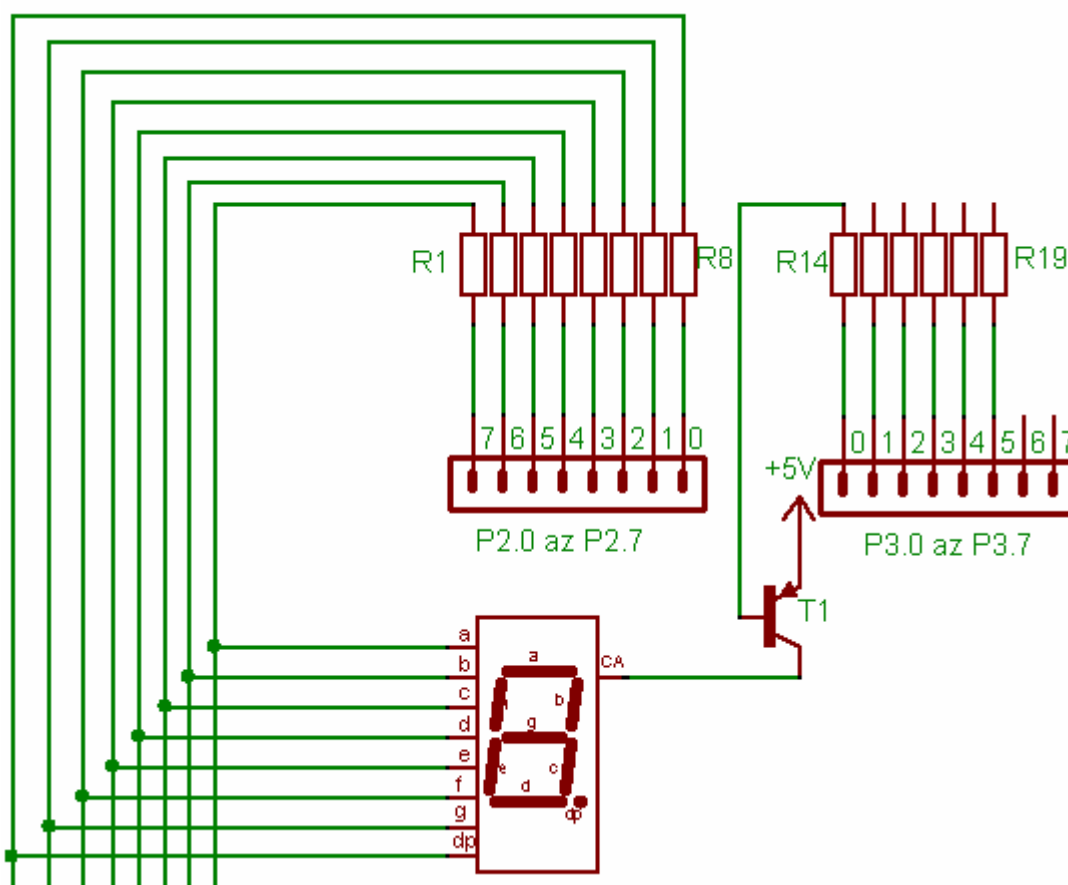
Segmentový displej se skládá ze segmentů (diod), které se dají nezávisle na sobě ovládat. Tedy rozsvěcet a zhasínat. V praxi se tyto segmentové displeje používají například v počítadlech, hodinách či při signalizaci teploty zařízení. Použití těchto displejů je všestranné. Mezi jejich výhody patří nízká pořizovací cena, jednoduché programování, dobrá viditelnost při zobrazování a minimální poruchovost i za extrémních teplotních podmínek.

4.1.1.1 Připojování segmentového displeje

Zobrazovací prvky lze připojit k mikropočítači pomocí řídicích obvodů nebo přímo. Jako řídicí obvody se zpravidla používají dekodéry. V případě desky VDT2 bylo

zvoleno přímé zapojení. Při tomto způsobu zapojení lze jednotlivé segmentové displeje připojit přímo na výstupy mikroprocesoru. Pokud by bylo nutné připojit velké množství sedmisegmentových displejů, bylo by potřeba také velké množství volných pinů na procesoru. Z toho důvodu bylo zvoleno takzvané multiplexní řízení segmentových displejů. Aby byly rozsvíceny všechny sedmisegmentové displeje najednou, s různým obsahem dat, je nutné mezi sebou přepínat jednotlivé sedmisegmentové displeje takovou rychlostí (alespoň 25x za sekundu), aby lidské oko nepostřehlo rozdíl mezi zhasnutým a rozsvíceným displejem. Multiplexní řízení má i jednu nevýhodu, kterou je velká časová náročnost obsluhy segmentových displejů mikropočítačem.

K zapojení sedmisegmentových zobrazovačů bylo použito osm výstupů mikroprocesoru na řízení katod, jež jsou umístěny na portu P2.0 až P2.7 a šest výstupů na řízení anod, jež jsou připojeny na portu P3.0 až P3.5. Na anody jsou připojeny tranzistory, které umožňují sedmisegmentovému displeji proudový odběr až do výše 100 mA. Tento proud by neměl být překročen, protože každý sedmisegmentový displej má podle katalogu [10] maximální odběr 80 mA (10 mA na diodu).



Obr.15: Zapojení jednoho sedmisegmentového displeje

Výpočet:

V katalogu [10] jsou uvedeny hodnoty, které jsou důležité k výpočtu předřadných odporů. Použité segmentové displeje mají požadovaný proud pro svítící LED-diodu $I_f=10\text{ mA}$ a napětí $U_v=3,8\text{ V}$ při napájecím napětí $V_{dd}=5\text{ V}$. Při výpočtu hodnot odporů R_1 až R_8 a jejich jmenovitého zatížení. je nutno brát v úvahu úbytek napětí při otevřeném tranzistoru T_1 ($U_{ec}=0,2\text{ V}$) a úbytek napětí na tranzistoru uvnitř portu (přibližně $U_{ecp}=0,2\text{ V}$) (obr.15).

$I_f=10\text{ mA}$; $U_v=3,8\text{ V}$; $V_{dd}=5\text{ V}$; $U_{ec}=0,2\text{ V}$; $U_{ecp}=0,2\text{ V}$; $R=?$

$R_1=(V_{dd}-U_v-U_{ec}-U_{ecp})/I_f = (5-3,8-0,2-0,2)/0,01 = \underline{80\ \Omega}$ - nejbližší hodnota v katalogu je **$82\ \Omega$** .

Důležitý je i výpočet jmenovité zatížení odporů pro správnou volbu pouzdra.

$P=U \cdot I = (5-3,8-0,2-0,2) \cdot 0,01 = \underline{8\text{ mW}}$

Mohou být použity odpory SMD - velikosti 0805, které mají jmenovité zatížení $0,125\text{ W}$ při 70°C .

Výpočet:

Jeden sedmissegmentový displej má při sepnutí všech diod odběr 80 mA . Přívod segmentového displeje je dimenzován na hodnotu 100 mA . Při překročení této hodnoty hrozí zničení tranzistoru PNP BC 557C. Pro výpočet odporu (jenž je umístěn mezi bází tranzistoru a konektorem mikroprocesoru) a jeho jmenovitého zatížení byly použity hodnoty z katalogu elektrotechnických součástek [10] a schéma zapojení (obr.15). V katalogu má tranzistor PNP BC 557C uvedeny hodnoty: $I_c=100\text{ mA}$ a $h_{fe}=420-800$. Napětí zdroje je $V_{dd}=5\text{ V}$. Při výpočtu hodnot odporů R_{14} až R_{19} a jejich jmenovitého zatížení je uvažován úbytek napětí na tranzistoru T_1 mezi emitorem a bází $U_{eb}=0,7\text{ V}$ a úbytek na tranzistoru, jenž je umístěn v portu $U_{ecp}=0,2\text{ V}$.

$U_{dd}=5\text{ V}$; $h_{fe}=420 - 800$; $I_c=100\text{ mA}$; $U_{eb}=0,7\text{ V}$; $U_{ecp}=0,2\text{ V}$; $R_{14}=?$; $P=?$

$h_{fe} = I_c/I_b \Rightarrow I_b = I_c/h_{fe} = 100/420 = \underline{0,238\text{ mA}}$

$R_{14}=(V_{dd}-U_{eb}-U_{ecp})/I_b = 4,1/0,000238 = \underline{17,23\text{ k}\Omega}$ - nejbližší hodnota v katalogu je **$18\text{ k}\Omega$** .

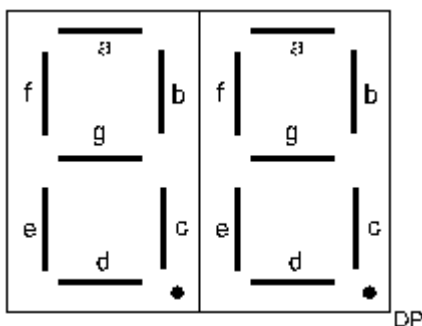
Výpočet jmenovitého zatížení odporů.

$$P = (V_{dd} - U_{eb} - U_{ecp}) \cdot I_b = 4,1 \cdot 0,000238 = \underline{\underline{0,9 \text{ mW}}}$$

Může být použit SMD-rezistor o velikosti 0805, který má jmenovité zatížení 0,125 W při 70°C.

4.1.1.2 Programování segmentového displeje

Každý sedmisegmentový displej má v sobě integrováno osm diod. Každá dioda má přiřazeno vlastní označení (písmenko v abecedě) (obr.16). Pokud bude požadováno zobrazení čísla „1“, musí programátor na aktivní port poslat kód, kde na pozicích diod (b,c) bude logická úroveň o hodnotě 0. Například: 10011111.



Obr.16: Popis jednotlivých segmentů

Každý bit na portu P2.0 až P2.7 určuje jednu diodu na segmentovém displeji.

Na katody displejů bude posílána bitová kombinace jedniček a nul; kde jednička deaktivuje a nula aktivuje daný segment displeje. Bitová kombinace zasílaná na anody segmentových displejů v logické úrovni 0 aktivuje displej, 1 displej deaktivuje.

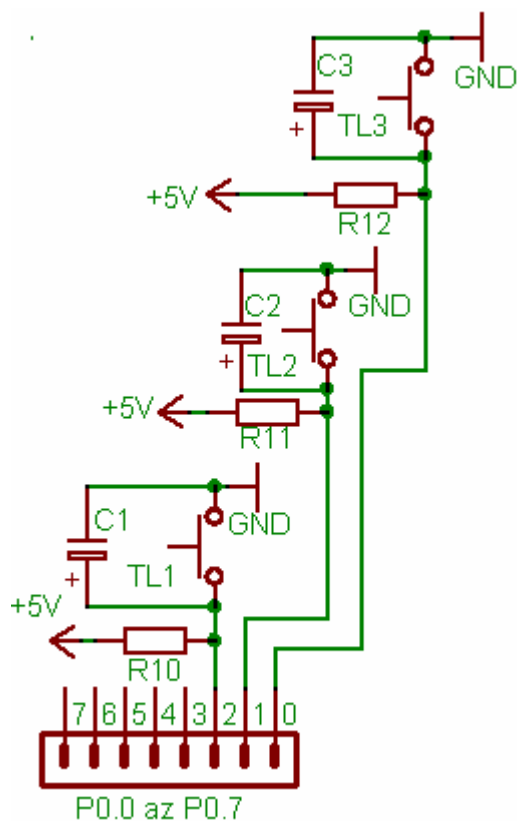
4.1.2 Tlačítka na desce VDT2

Na desce VDT2 jsou umístěna tři tlačítka, která slouží k naprogramovanému ovládání (segmentového displeje reproduktoru nebo obvodu RTC). Tlačítka mají v katalogu [10] označení P-DT6SW a byla zvolena s ohledem na svou velikost.

4.1.2.1 Připojování tlačítek

Připojování tlačítek je velmi podobné jako připojování klávesnice. Jediný rozdíl je ten, že každé tlačítko je připojeno na vlastní pin na portu. Pokud by tlačítek mělo být více než je volných pinů na portech, je vhodnější zvolit pro ovládání klávesnici nebo maticové zapojení tlačítek. Tlačítka je možné připojit dvěma způsoby, a to metodami,

které se nazývají tzv. kladná a záporná logika. Jak je patrné z obr.17, jedná se o “negativní” neboli zápornou logiku. Je-li tlačítko sepnuto, objeví se na vstupu mikroprocesoru logická úroveň o hodnotě 0. Z uvedeného vyplývá, že pokud bude tlačítko rozepnuto, objeví se na vstupu mikroprocesoru logická úroveň 1. Jednotlivá tlačítka jsou připojena přímo na port mikroprocesoru a to z dostatečného množství pinů na portu. Tlačítka jsou zapojena na port P0.0 až P0.2. Protože port P0 nemá zdvihací odpory, musely být umístěny mezi jednotlivé piny a napájecí napětí. Pokud by zdvihací odpory nebyly použity, zapojení tlačítek by nefungovalo. Zdvihací odpory zajišťují na vstupu mikroprocesoru logickou úroveň o hodnotě 5V. Stiskem tlačítka dojde k uzemnění napětí a nastane tak na procesoru stav logické úrovně 0. K tlačítkům byly zvoleny do paralelní kombinace kondenzátory, které by měly zabezpečit ošetření kontaktů proti zákmitu. Zakmitávání kontaktů se dá řešit i softwarově. Hodnota kondenzátorů byla zvolena na velikost 1 μF .



Obr.17: Zdvihací odpory s připojenými tlačítky na portu P0

Výpočet:

Výpočet zdvihacích odporů a jejich jmenovitého zatížení. Zdvihací odpory se zpravidla volí v rozmezí hodnot 1000 až 5000 Ω . Napájecí napětí je $V_{dd}=5\text{ V}$, volený proud

$I=1\text{ mA}$. Úbytek napětí mezi emitorem a kolektorem tranzistoru umístěným uvnitř portu je $U_{ecp}=0,2\text{ V}$. Nyní dojde k výpočtu odporů R_{10} až R_{12} a jejich jmenovitého zatížení P_{zd} .

$V_{dd}=5\text{ V}$; $I=1\text{ mA}$; $U_{ecp}=0,2\text{ V}$; R_{10} až $R_{12}=?$; $P_{zd}=?$

$$R_{zd}=(V_{dd}-U_{ecp})/I = 4,8/0,001 = \underline{4800\ \Omega} \Rightarrow \underline{4700\ \Omega}$$

$$P_{zd}=(V_{dd}-U_{ecp})\cdot I = 4,8\cdot 0,001 = \underline{0,48\text{ mW}}$$

Hodnota zdvihacího odporu je $4k7$ a jmenovité zatížení je $0,48\text{ mW} \Rightarrow$ může být použit SMD odpor velikosti 0805.

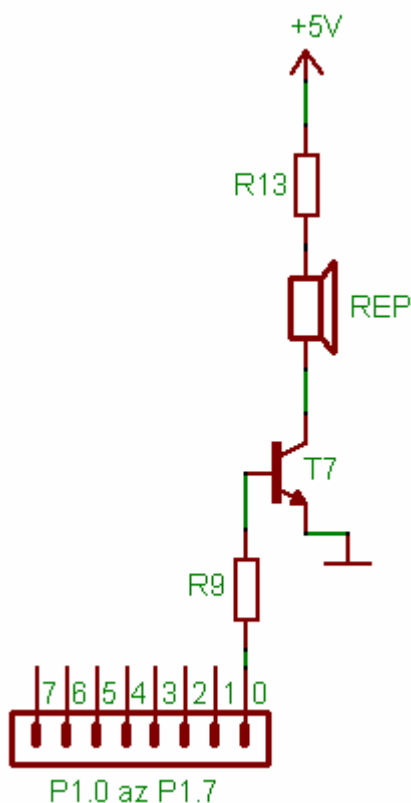
4.1.3 Reproduktor na desce VDT2

Reproduktor slouží v průmyslu převážně jako bezpečnostní signalizační zařízení. Například signalizace odpojení ovládacích obvodů, zabezpečujících prvků, či oznamování výpadků ovládacího napětí. Reproductory jsou používány i v běžném životě. Například mobilní telefony, budíky, rádia a mnoho dalších přístrojů a zařízení.

Bylo řečeno [11], že reproductory jsou elektro-akustické měniče, tj. zařízení, která přeměňují elektrický proud na zvuk. Obvykle se skládají z membrány, vyrobené z plastu nebo papíru, a z elektromagnetu, do něhož je přiváděn vstupní signál. Zvláštním případem reproduktoru jsou sluchátka. Kromě principu elektromagnetu se v některých případech používá k rozkmitání membrány i piezoelektrický jev.

4.1.3.1 Připojování reproduktoru

Reproduktor je připojen na port P1.0 (obr.18). Reproduktor je spínán pomocí tranzistoru T7, jenž je připojen bází přes odpor R_9 na port procesoru. Druhý pin reproduktoru je přes odpor R_{13} připojen na napájecí napětí. Velikost odporu R_{13} určuje hlasitost reproduktoru. Pokud by měla být hlasitost reproduktoru měnitelná, musel by být namísto odporu R_{13} použit potenciometr.



Obr.18: Zapojení reproduktoru

Výpočet:

Výpočet odporů R13, R9 a jejich jmenovitého zatížení P13 a P9.

Jako spínací tranzistor pro obvod s reproduktorem byl zvolen NPN BC 546B, který má činitel zesílení $H_{fe}=200$ až 450 a maximální proud kolektorem $I_c=0,1$ A. Úbytek napětí na tranzistoru T7 mezi emitorem a kolektorem při sepnutém tranzistoru je $U_{ec}=0,2$ V a úbytek napětí mezi emitorem a bází je $U_{eb}=0,7$ V. Proud do báze byl zvolen $I_b=1$ mA. Úbytek na tranzistoru uvnitř portu je $U_{ecp}=0,2$ V. Miniaturní reproduktor k tranzistoru byl vybrán LP-21008. Jeho hodnoty z katalogu [10] jsou výkon $P_r=0,1$ W a impedance $Z_r=8 \Omega$.

$Z_r=8\Omega$; $P_r=0,1$ W; $U_{ec}=0,2$ V; $U_{eb}=0,7$ V; $U_{ecp}=0,2$ V; $V_{dd}=+5$ V; $I_c=0,1$ A

$R_{13}=?$; $R_9=?$; $P_{13}=?$ a $P_9=?$ (R_c = odpor sériové kombinace R13 a Z_r)

$$R_c = (V_{dd} - U_{ec}) / I_c = (5 - 0,2) / 0,1 = \underline{\underline{48 \Omega}}$$

$$R_{13} = R_c - Z_r = 48 - 8 = \underline{\underline{40 \Omega}} \text{ - nejvyšší bližší je } \underline{\underline{48 \Omega}}$$

$$P_{13} = (V_{dd} - U_{ec}) \cdot I_c = 4,8 \cdot 0,1 = \underline{\underline{0,48 \text{ W}}} \Rightarrow \text{musí být použit metalizovaný odpor s jmenovitým zatížením minimálně } \underline{\underline{0,6 \text{ W}}}$$

$R9 = (V_{dd} - U_{eb} - U_{ecp}) / I_b = (5 - 0,7 - 0,2) / 0,001 = \underline{4,1 \text{ k}\Omega} \Rightarrow$ z katalogu vybrán **4,7 k Ω**

$P9 = (V_{dd} - U_{eb} - U_{ecp}) * I_b = 4,1 * 0,001 = \underline{4,1 \text{ mW}} \Rightarrow$ lze použít SMD odpor velikosti 0805.

4.1.3.2 Programování reproduktoru.

K ovládání reproduktoru z portu procesoru je použit obdélníkový signál s naprogramovanou časovou periodou. Perioda signálu určuje výšku tónu. Počet period signálu určuje délku tónu. Během jedné periody může být vyslán stejný počet jedniček i nul (logických úrovní) a tím vzniká symetrický signál. Samozřejmě je možné vytvořit také asymetrický signál, který je tvořen rozdílným počtem jedniček a nul během jedné periody.

4.1.4 Obvod reálného času RTC (real time clock)

Obvody RTC jsou používány například v aplikacích budíků. Nebo v aplikacích, kdy je potřeba provádět přesné spínání výkonové zátěže.

Výběr obvodu RTC pro desku VDT2 se odvíjel od níže uvedených požadavků. Bylo požadováno připojení pomocí rozhraní SPI, přítomnost funkce budíku, jednoduché zálohování baterií a možnost dobíjení baterie. Jako vhodný typ byl vybrán obvod DS1305 firmy Dallas Semiconductor.

Obvod DS 1305 [12] a jeho vlastnosti:

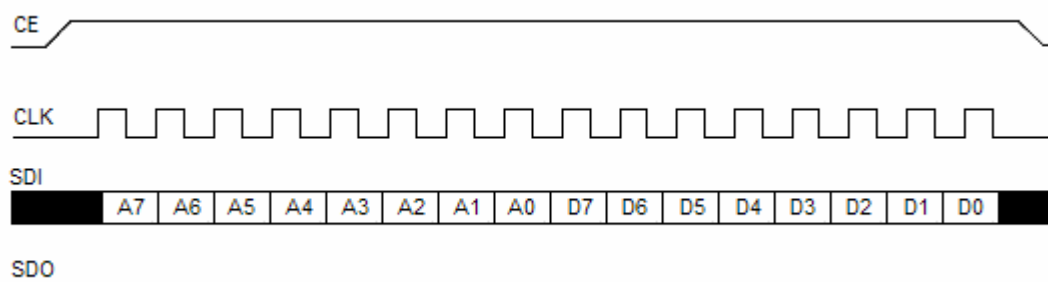
- Uchovává čas ve formátu DD/MM/RR, HH.MM:SS.
- Má v sobě zabudovanou informaci o dnu v týdnu.
- Rozpozná přestupný rok.
- Obsahuje dva budíky s vyvedenými signály podle požadavku přerušení.
- Obsahuje 96 bytů paměti RAM pro libovolné použití.
- Paměť je zálohovaná baterií.
- Hodinový kmitočet je odvozován od vnějšího krystalu 32768 kHz.
- Je možné připojení přes sběrnici SPI nebo 3-wire.
- Obvod je kalibrován do roku 2100.
- Hodnota napájecího napětí je 5 V.

4.1.4.1 Připojování RTC

Obvod je napájen +5V na pinu VCC. Napájecí napětí je přivedeno rovněž na piny VCCIF a SERMODE. Jako zálohu pro svou paměť používá obvod baterii, jež je připojena na piny VCC2 a VBAT. Baterie musí být uzemněna s piny GND, které jsou umístěny v obvodu RTC. Na piny X1 a X2 je připojen krystal, který řídí celý obvod. Komunikační piny, které jsou připojeny na port P1.4 (SDO) P1.5 (SCLK) P1.6 (SDI) a P1.7 (CE), slouží ke komunikaci mezi procesorem a obvodem reálného času pomocí sběrnice SPI.

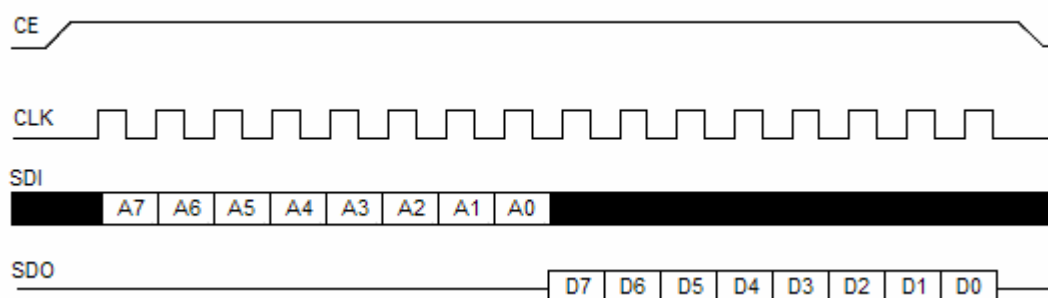
4.1.4.2 Programování RTC

Při programování obvodu RTC byla zvolna softwarová komunikace po sběrnici SPI. To znamená, že všechna data, adresy a potvrzující bity musí být řízeny softwarově.



Obr.19: Obr. Zápis dat do paměti obvodu RTC.

Při softwarovém řízení zápisu dat (Obr. 19). do obvodu RTC po sběrnici SPI musí být nejdříve přiveden signál CE a SDO do logické úrovně 1 následně je vyslán signál CLK a zapsán daný bit SDI. Takto jsou zapsána všechna potřebná data do obvodu RTC. Po skončení posílání dat je signál CE uveden do logické úrovně „0“. Podobným způsobem se i data z obvodu RTC získávají (Obr. 20).



Obr.20: Čtení dat z paměti obvodu RTC

System čtení dat z paměti obvodu RTC je zobrazen na obr.20. Nejdříve je přiveden signál CE do logické úrovně 1. Následně je provedeno zaslání signálu CLK společně se signálem SDI. Po zaslání všech bitů adresy bude s hodinovým signálem čten signál SDO, který obsahuje čtené informace. Po dokončení operace je nutné přivést signál CE do logické úrovně 0.

K programování obvodu RTC je nutné znát adresy, na které se data zapisují, a adresy, z kterých se data čtou. K tomuto slouží následující obrázek (obr.38).

Hexadecimální adresa		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Rozsah
READ	WRITE									
00H	80H	0	Desítky sekund			Sekundy			00-59	
01H	81H	0	Desítky minut			Minuty			00-59	
02H	82H	0	12	P	Desítky hodin	Hodiny			01-12 + P/A	
				A					00-23	
				24						10
03H	83H	0	0	0	0	Den			1-7	
04H	84H	0	0	Desítky data		Datum			1-31	
05H	85H	0	0	Desítky měsíců		Měsíc			01-12	
06H	86H	Desítky let				Rok			00-99	
-	-	Alarm 0							-	
07H	87H	M	Desítky sekund alarmu			Sekundy alarmu			00-59	
08H	88H	M	Desítky minut alarmu			Minuty alarmu			00-59	
09H	89H	M	12	P	Desítky hodin	Hodinový alarm			01-12 + P/A	
				A					00-23	
				24						10
0AH	8AH	M	0	0	0	Denní alarm			01-07	
-	-	Alarm 1							-	
0BH	8BH	M	Desítky sekund alarmu			Sekundy alarmu			00-59	
0CH	8CH	M	Desítky minut alarmu			Minuty alarmu			00-59	
0DH	8DH	M	12	P	Desítky hodin	Hodinový alarm			01-12 + P/A	
				A					00-23	
				24						10
0EH	8EH	M	0	0	0	Denní alarm			01-07	
0FH	8FH	Kontrolní registr							-	
10H	90H	Stavový registr							-	
11H	91H	Nabíjecí registr							-	
12-1FH	92-9FH	Rezervováno							-	
20-7FH	A0-FFH	96 Bytů uživatelské RAM							00-FF	

Obr.21: Funkce obvodu RTC s přiřazenými adresami

Při prvním použití obvodu je důležité aktivovat daný obvod. K tomuto účelu slouží kontrolní registr (obr.22).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EOSC	WP	0	0	0	INTCN	AIE1	AIE0

Obr.22: Kontrolní registr

- EOSC- při přivedení logické úrovně 0 se aktivuje oscilátor; při přivedení logické úrovně o hodnotě 1 se oscilátor vypne a obvod, pokud je z baterie zálohově napájen, sníží svůj proudový odběr na hodnotu 100 nA
- WP- bude-li zapisováno do paměti RAM obvodu RTC, musí tento bit být v logické úrovni 0.
- INTCN-tento bit v logické úrovni 1 povoluje dva denní alarmy a přerušení výstupního pinu.
- AIE0- v logické úrovni 1 povoluje přerušení od alarmu 0, v logické úrovni 0 daný signál přerušení není vyslán.
- AIE1-v logické úrovni 1 povoluje přerušení od alarmu 1, v logické úrovni 0 daný signál přerušení není vyslán.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	IRQF1	IRQF0

Obr.23: Stavový registr

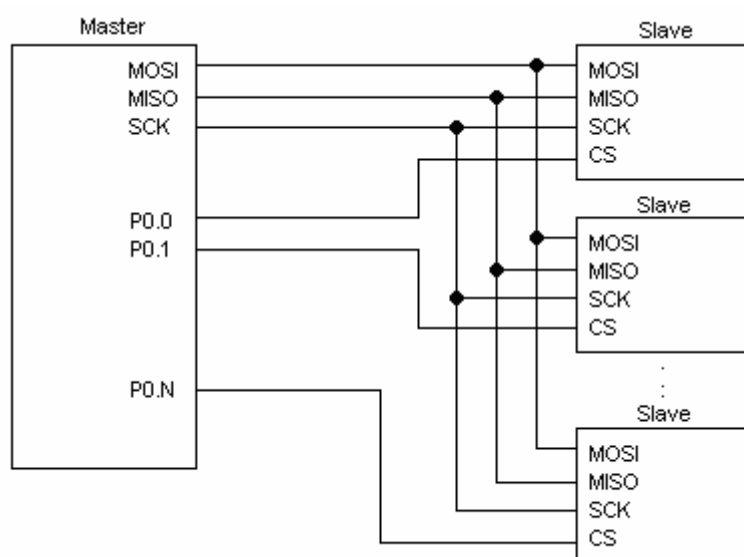
- IRQF0-pokud je tento bit v logické úrovni 1, dochází k dotazování bitu AIE0 v alarmu 0, pokud tento bit je v logické 1, bude pin INT0 nastaven do logické úrovně 0.
- IRQF1- pokud je tento bit v logické úrovni 1, dochází k dotazování alarmu 1. Přerušení může být generováno pro INT0 a INT1. Volbu mezi těmito stavy určuje bit INTCN v kontrolním registru. Pokud bude IRQF1, AIE1 a INTCN v logické úrovni 1 INT1 bude v logické úrovni 0. Pokud však bude IRQF1, AIE1 v logické úrovni 1 a INTCN v logické úrovni 0 INT0, bude v logické úrovni 0.

4.1.4.3 Funkce Rozhraní SPI

Bylo řečeno [13], že rozhraní SPI je především určeno pro připojení vnějších pamětí, A/D převodníků a dalších obvodů k mikrokontroléru, případně pro vzájemnou komunikaci mezi mikrokontroléry. U některých klonů mikrokontrolérů je rozhraní SPI využíváno i pro programování jejich vnitřní paměti.

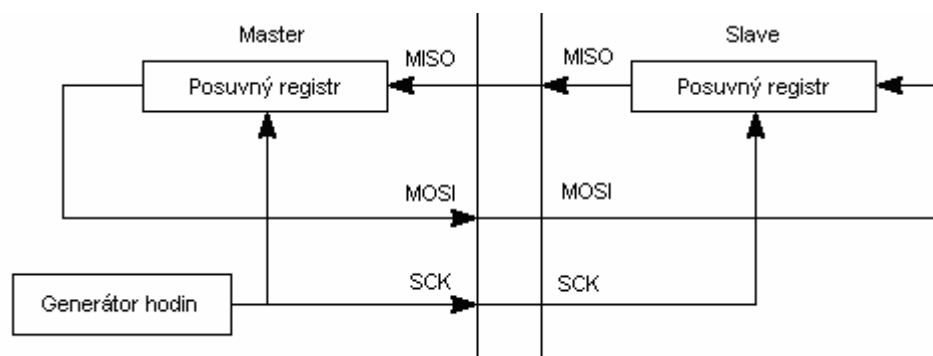
Na sběrnici mohou být zapojeny dva nebo více obvodů. Jeden z obvodů, obvykle řídí mikrokontrolér, je typu Master, ostatní jsou typu Slave. Komunikace probíhá pomocí 4 vodičů:

- Datový výstup MOSI (Master Out, Slave In) je připojen na vstupy MOSI všech obvodů Slave.
- Datový výstup MISO (Master In, Slave Out) obvodu Master je propojen s výstupy MISO všech obvodů Slave.
- Výstup hodinového signálu SCK je připojen na vstup SCK všech obvodů Slave.
- CS(Chip Select) je vývod pro výběr obvodu. Je-li nastaven v neaktivní úrovni, je příslušný Slave obvod odpojený od sběrnice, jeho vstupy jsou ve vysoko-impedančním stavu. Je-li obvodem Master mikrokontrolér, bývají vstupy připojeny na některou z jeho bran. Tímto způsobem lze snadno vybírat obvod, s nímž má být vedena komunikace.



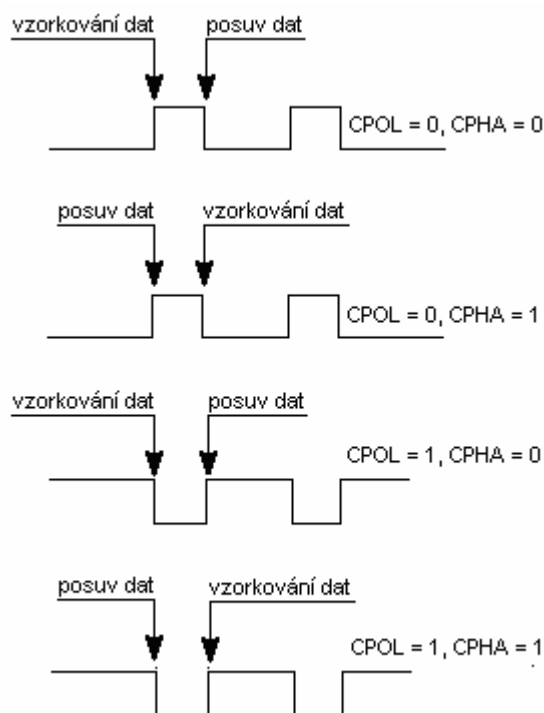
Obr.24: Koncepce systému se sběrnici SPI

Přenosy po sběrnici SPI probíhají vždy mezi obvodem Master a obvodem Slave. Oba obvody obsahují posuvné registry, které jsou propojeny tak, jak je schématicky naznačeno na obr. 25.



Obr.25: Propojení obvodů Master a Slave

Obvod Master generuje hodinový signál, který řídí posouvání obou posuvných registrů. Klidová úroveň signálu SCK a vztah mezi datovým a hodinovým signálem je dán parametry CPOL a CPHA (obr.26). Pokud je rozhraní SPI realizováno specializovaným řadičem, je obvykle možné tyto parametry v řadiči nastavit. Je-li rozhraní SPI realizováno programově, musí být okamžiky změny úrovně datových a hodinových signálů zvoleny tak, aby přijímací obvod vzorkoval platná data.

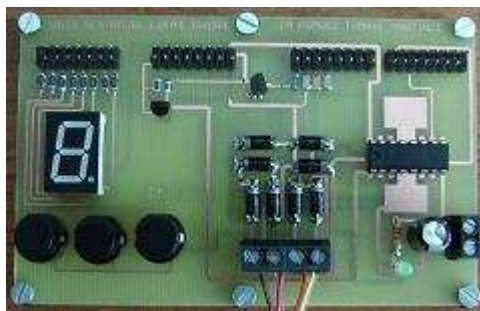


Obr.26: Význam parametrů CPOL a CPHA

Napěťové úrovně jednotlivých signálů rozhraní SPI jsou dané použitou technologií. Maximální frekvence hodinového signálu je 2 MHz.

5. Třetí výuková deska VDT3

Tato výuková deska vznikla za účelem seznámení studentů s programováním a zapojováním krokových motorků, v kombinaci se zobrazovačem a jednotlivými tlačítky. Výuková deska byla z ohledem na cenu, velikost a proudové zatížení zkonstruována s obvodem, který má v sobě integrovány dva H-můstky. Jedná se o obvod L293D. Elektrotechnické schéma desky VDT3 je v příloze č.4 tohoto dokumentu.

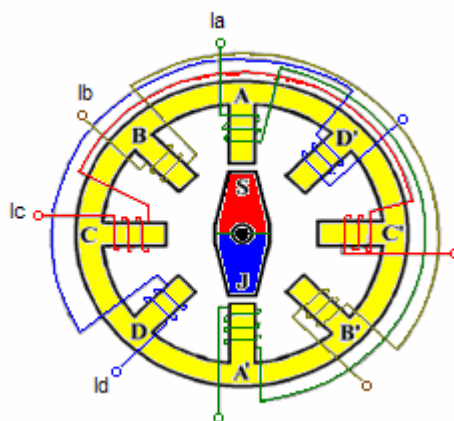


Obr.27: Deska VDT3

5.1 Teorie o Krokových motorech

V této aplikaci bylo požadováno především co nejlevnější řešení pro možnosti výuky. Jako ideální řešení pro tento případ byl použit motorek ze staré disketové mechaniky. Tento motorek má vyvedené pouze celé cívky, tzn. že pro jeho ovládání jsou využity všechny čtyři jeho vývody. Bylo provedeno měření vývodů a zjištěn odpor vždy mezi dvěma páry vodičů. Z toho vyplývá, že u tohoto motorku je pouze jedna varianta řízení a to bipolární pomocí dvou H-můstek.

Bylo řečeno [14], že se krokové motory dělí podle provedení na unipolární a bipolární. Bipolární krokové motory mají rotor tvořený permanentním magnetem. Každé fázi odpovídá jedno vinutí, které je podle konstrukce příslušného motoru navinuto na jednom či více pólových párech, a oba jeho konce jsou vyvedeny (obr. 28). Počet vývodů motoru je tak dvojnásobkem počtu fází. Ke změně směru proudu a jím vyvolaného magnetického toku je nezbytné změnit polaritu napětí na příslušné fázi. Z tohoto důvodu je toto uspořádání označováno jako **bipolární**.



Obr.28: Řez krokovým motorem s permanentním magnetem

U unipolárních krokových motorů je rotor z magneticky měkkého materiálu a všechny cívky mají jeden z vývodů zapojen na společný přívodní vodič.

Body propojení jednotlivých fází motorku jsou nejčastěji již přímo uvnitř spojeny dohromady. Počet vývodů motorku je pak roven dvojnásobku počtu fází zvětšenému o jednu. Je zřejmé, že výhodou tohoto uspořádání je jednoduchost. Není nutné měnit polaritu napětí na cívce, ale postačí spínat její jednu či druhou polovinu proti zemi. Vzhledem k tomu se toto uspořádání obvykle označuje jako **unipolární**. Jelikož motorek na (obr. 28) má celkem čtyři spínané vývody, lze se často setkat s tím, že je ne tak docela správně označován jako čtyřfázový. Stojí také za zmínku, že vinutí jedné fáze nejsou nezávislá, ale jsou propojená a navinutá na stejných pólových párech. Fungují tedy podobně jako autotransformátor a v důsledku toho se mezi kolektorem a emitorem spínacích tranzistorů mohou objevit záporné napěťové špičky. Někdy bývá proto doporučováno použít další čtveřici ochranných diod zapojených tak, že jejich katoda je spojena s kolektorem spínacího tranzistoru a anoda se zemí. Zvláště u motorků menších výkonů, jejichž proudový odběr se pohybuje ve stovkách mA či jednotkách A, je však nejvhodnějším a nejlevnějším řešením výkonové části použití integrovaných obvodů jako ULN2803A nebo ULN2064B apod., které obsahují skupinu spínačů v Darlingtonově uspořádání spolu s ochrannými diodami a bázovými odpory vhodně zvolenými tak, aby mohly být buzeny přímo z výstupů obvodů TTL. Výkonový obvod pak je velmi jednoduchý a levný.

5.1.1 Výhody a nevýhody mezi unipolárním a bipolárním zapojováním

Hlavní výhodou unipolárních motorů je jednoduchost jejich výkonových obvodů, zatímco ovládací obvody bipolárních motorů vyžadují pro každou fázi jeden H-můstek pro změnu polarity. Z tohoto důvodu byla unipolárnímu uspořádání dávána po dlouhou dobu přednost a ve starší literatuře se lze běžně setkat s názorem, že toto uspořádání je vhodnější a obecně preferované. V současné době však je přinejmenším pro motory s menším výkonem snadno dostupná řada monolitických integrovaných obvodů obsahujících H-můstky (obvykle dva). Z hlediska počtu součástek a složitosti i ceny výkonových budičů pak již mezi bipolárními a unipolárními motory není podstatný rozdíl. U unipolárních motorů se pak projeví spíše nevýhody. Množství měděného vinutí, které lze vtěsnat do motoru určité velikosti, je omezeno a vinutí unipolárních motorů, které mají ve stejném prostoru místo jednoho vinutí dvě, proto musí být provedeno vodičem menšího průřezu. Důsledkem je větší odpor a tedy při stejném napájecím napětí menší proud a menší moment na hřídeli motoru. Obojí lze zvýšit použitím vyššího napájecího napětí. Nepřekročitelným omezením je však vzrůstající výkonová ztráta, která je při stejném proudu a větším odporu samozřejmě úměrně zvětšuje a způsobuje zahřívání vinutí motoru. Při stejné výkonové ztrátě na motoru tak bipolární motory vytvářejí asi o 40% větší moment či z jiného pohledu, bipolární motor se stejnou výkonovou zatížitelností bude menší a s menší výkonovou ztrátou než odpovídající unipolární. Z těchto důvodů se v současnosti motory s bipolárním uspořádáním vinutí jeví obvykle jako výhodnější.

5.2 H-můstek

Obvod který byl vybrán pro řízení krokového motorku na desce VDT3 má v sobě integrované dva H-můstky. Jedná se o obvod L293D firmy Texas Instrument.

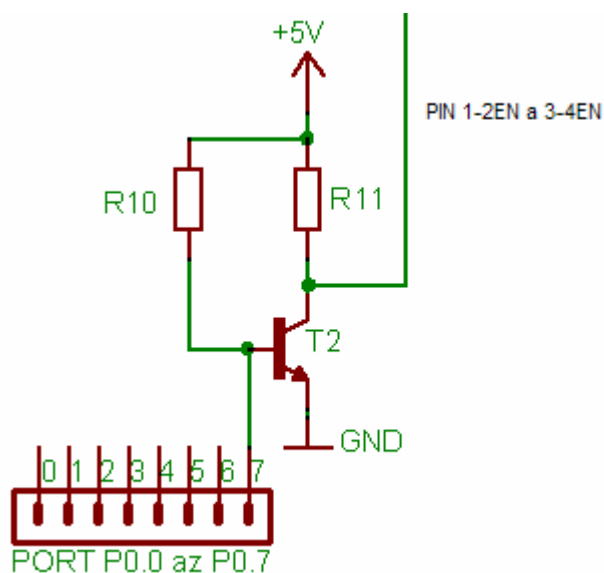
Technická specifikace obvodu L293D [15]

- Široký napájecí rozsah výkonové části 4,5 až 36 V
- Oddělená vstupní logika od napájení
- Obsahuje tepelnou ochranu před přetížením
- Vysokou šumovou imunitu vstupů
- Funkčností podobný obvodu L293
- Výstupní proud na kanál 600 mA

- Špičkový výstupní proud na kanál 1,2 A
- Integrované výstupní zhášecí diody pro induktivní přechodové napětí
- Řídící napájecí logika je +5V

5.2.1 Připojování L293D

Obvod L293 je připojen následovně. Vstupy určené pro řízení cívek 1A až 4A jsou přivedeny na piny P1.7 až P1.4. Výstupy určené pro cívky motoru 1Y až 4Y jsou vyvedeny na svorkovnici, ke které lze připojit krokový motorek. Piny 1-2EN a 3-4EN, které povolují sepnutí výkonové části obvodu, jsou připojeny na port P0.7 tak, že při zapnutí procesoru nemůže být sepnuta výkonová zátěž, aby nedošlo k úrazu elektrickým proudem. Ochranná část obr.29.



Obr.29: Povolení sepnutí výkonového obvodu

Odpor R11 byl zvolen z datasheetu k obvodu L293D [15], má hodnotu 10 kΩ a odpor R10 má hodnotu zvyšovacího odporu 4,7 kΩ. Tranzistor T2, který v tomto případě určuje volbu mezi aktivací a deaktivací pinů 1-2EN a 3-4EN, byl vybrán typ BC547B. Bude-li na výstupu mikroprocesoru logická úroveň 0, potom bude mít obvod L293D sepnutou výkonovou část. Pokud bude ovšem na výstupu procesoru logická úroveň 1, bude výkonová část obvodem L293D blokována.

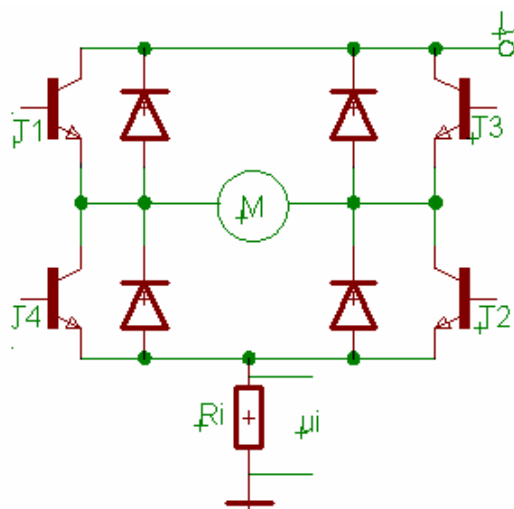
Jako vstupní napětí pro výkonovou část je možné použít napětí ze stabilizovaného zdroje stejnosměrného napětí až do velikosti 24 V. Přítomnost výkonové části signalizuje zelená dioda u vstupních svorek zdroje. Krokovému

motorku, který je řízen touto deskou, stačí napájecí napětí přibližně 5V. Důležité tedy je, aby napájecí napětí nepřekračovalo hodnotu napájecího napětí motorku, aby nemohlo dojít k jeho zničení. Při přepínání jednotlivých cívek do protipohybu může nastat situace, kdy vznikne na cívkách motorku indukční přechodové napětí. Díky tomuto napětí by mohlo dojít ke zničení řídicího obvodu L293D. Přestože má tento obvod v sobě zabudované ochranné diody, byly mezi obvod a konektory pro krokový motorek umístěny rychlé diody, které mají v případě indukčního přechodového napětí zabránit zničení obvodu.

5.2.2 Řízení obvodu L293D

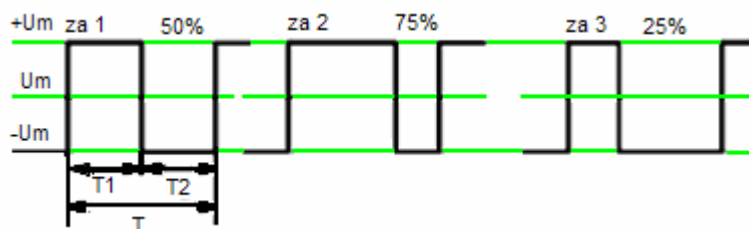
Bylo řečeno [14], že H-můstek je označení pro charakteristicky uspořádané tranzistorové spínače.

Při sepnutých tranzistorech T1 a T2 (obr.30) polarita napětí na motoru odpovídá jedné polaritě, při sepnutých tranzistorech T3 a T4 je polarita napětí na kotvě opačná. V můstku mohou být podle okolností užity bipolární tranzistory i tranzistory MOSFET či IGBT. Malý odpor R_i slouží ke snímání velikosti proudu procházejícího můstkem pro účely proudové zpětné vazby nebo v jednodušším případě alespoň nadproudové ochrany výkonových tranzistorů. U některých modernějších integrovaných H-můstků (např. LMD18200 firmy National Semiconductor) se lze setkat s využitím modifikované struktury spínacích tranzistorů MOSFET umožňující získání informace o proudu i bez měřicího odporu. Zvláště při větších výkonech bývá snímání proudu často řešeno pomocí Hallových sond. Výhodou H-můstku je, že umožňuje reverzaci i brzdění krokového motoru.



Obr.30: Zapojení H-můstku

Řízení H můstku může probíhat dvěma základními způsoby. První způsob znázorňuje obr. 31.



Obr.31: Řízení H-můstku dvěma průběhy v protifázi

Během T_1 jsou sepnuty tranzistory T_1 a T_2 , po dobu T_2 pak T_3 a T_4 . Je-li označeno napětí o polaritě odpovídající řádku 1 (obr.30) jako kladné, je při různých poměrech T_1 a T_2 dosaženo průběhu jako na obr. 31. Jelikož platí $T_1 + T_2 = T$, lze pro střední hodnotu napětí přiváděného na kotvu motoru psát:

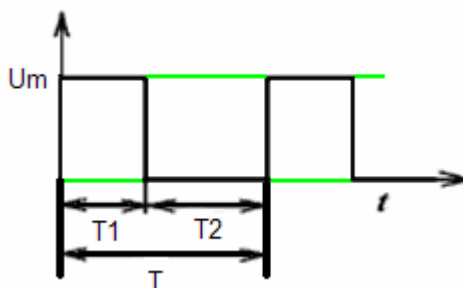
$$U_{aS} = (T_1 - T_2) U_m \quad T = 2(T_1 - 0,5) U_m.$$

Při rovnosti $T_1 = T_2$ je střední hodnota napětí přivedeného na kotvu motoru nulová a motor stojí. Při nerovnosti je podle znaménka rozdílu buď kladná nebo záporná, přičemž její velikost lze spojitě regulovat a motor se otáčí jedním či druhým směrem. Při tomto způsobu řízení je však efektivní hodnota napětí na kotvě motoru nenulová i při stojícím motoru a nulové střední hodnotě napětí. Veškerý protékající proud se pak mění v Jouleovo teplo. Tato varianta je proto vhodná pro rychlostní servomechanismy. Pro polohové servomechanismy je však použitelná pouze tehdy, když konstrukce motoru zajišťuje dostatečný odvod tepla i při stojícím motoru. Obvykle je proto v tomto případě výhodnější použít takový způsob řízení, kdy je pro otáčení v jednom směru trvale sepnut tranzistor T_1 a napětí na kotvě se reguluje pomocí tranzistoru T_4 . Pro otáčení v opačném směru je trvale sepnut tranzistor T_3 a PWM signál se přivádí na T_4 . Při stojícím motoru jsou všechny čtyři tranzistory rozepnuty a motor se tak ani nezahřívá, ani zbytečně nezatěžuje napájecí zdroj.

5.2.3 PWM (Pulse Width Modulation)

Bylo řečeno [14], že pulzně šířková modulace je proces, při kterém dochází k poměrné změně délky pulzů o konstantní frekvenci. To znamená, že perioda signálu je pořád stejná, rychlost otáčení motoru však měníme změnou času T_1 a T_2 od minima do maxima. Pokud bude potřeba točit motorem co nejrychleji, co nejvíce zvětšíme čas T_1

na úkor T2, pokud bude potřeba dosáhnout opačného jevu, bude muset být zvětšena velikost času T2 na úkor času T1. Velice důležité při používání PWM je používat frekvenci spínání tranzistorů nad slyšitelným pásmem lidského ucha. tj. alespoň 20 khz, aby nedocházelo k tomu, že motor bude zdrojem nepříjemných zvuků. Samozřejmě toho lze dosáhnout pouze tehdy, jsou-li spínací tranzistory postaveny tak, aby tak rychlého spínání byli schopny.



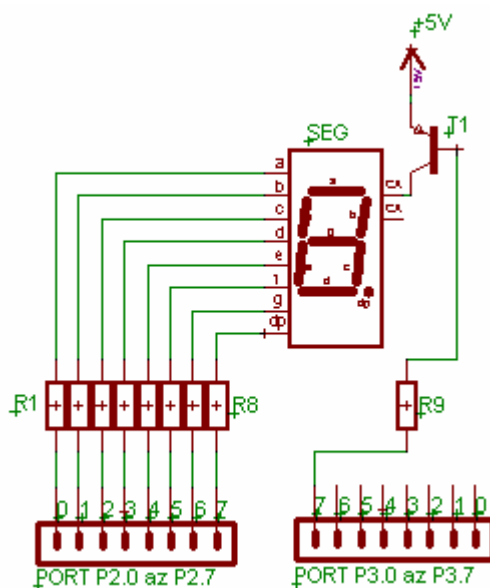
Obr.32: Pulzně šířková modulace

5.3 Ostatní součástky na desce VDT3

Ostatními součástkami na desce VDT3 jsou: tlačítka a segmentový displej. Obě uvedené součástky byly již v textu popsány v předchozích kapitolách. Proto jsou v této podkapitole uvedeny pouze informace o připojení a výpočtech.

5.3.1 Segmentový displej na desce VDT3

Jako segmentový displej na desce VDT3 byl zvolen zobrazovač s označením HD-A552RD zelené barvy. Z obrázku níže je patrné, že katody diod umístěných v pouzdře segmentového displeje jsou připojeny na port P2.0 až P2.7 a anoda segmentového displeje je připojena na pin portu P3.7.



Obr.33: Zapojení segmentové displeje na desce VDT3

Výpočet:

V katalogu [10] jsou uvedeny následující hodnoty, které jsou důležité k výpočtu předřadných odporů. Použitý segmentový displej má požadovaný proud pro svítící LED-diodu $I_f=10\text{ mA}$ a napětí $U_v=2,2\text{ V}$ při napájecím napětí $V_{dd}=5\text{ V}$. Při výpočtu hodnot odporů R1 až R8 (obr.33) a jejich jmenovitého zatížení je nutné brát v úvahu úbytek napětí při otevřeném tranzistoru T1 ($U_{ec}=0,2\text{ V}$) a úbytek napětí na tranzistoru uvnitř portu (přibližně $U_{ecp}=0,2\text{ V}$).

$I_f=10\text{ mA}$; $U_v=2,2\text{ V}$; $V_{dd}=5\text{ V}$; $U_{ec}=0,2\text{ V}$; $U_{ecp}=0,2\text{ V}$; $R=?$

$$R1=(V_{dd}-U_v-U_{ec}-U_{ecp})/I_f = (5-2,2-0,2-0,2)/0,01 = \underline{240\ \Omega}$$

nejbližší hodnota v katalogu je **240 Ω** .

Jmenovité zatížení pro správnou volbu pouzdra odporu je:

$$P=U \cdot I = (5-2,2-0,2-0,2) \cdot 0,01 = \underline{24\text{ mW}}$$

Z toho vyplývá, že mohou být použity odpory SMD-velikosti 0805, které mají jmenovité zatížení při 70°C $0,125\text{ W}$.

Výpočet:

Jeden sedmisegmentový displej má při sepnutí všech diod odběr 80 mA . Přívod segmentového displeje je dimenzován na hodnotu 100 mA . Při překročení této hodnoty hrozí zničení tranzistoru PNP BC 557C. Pro výpočet odporu (jenž je umístěn mezi bází

tranzistoru a konektorem mikroprocesoru) a jeho jmenovitého zatížení bude použit katalog elektrotechnických součástek [10] a schéma zapojení (obr.33). V katalogu jsou u tranzistoru PNP BC 557C uvedeny tyto hodnoty: $I_c=100\text{ mA}$ a $h_{fe}=420-800$. Napětí zdroje je $V_{dd}=5\text{ V}$. Výpočet hodnoty odporu R9 a jeho jmenovitého zatížení. Při výpočtu uvažujeme úbytek napětí na tranzistoru T1 mezi emitorem a bází $U_{eb}=0,7\text{ V}$ a úbytek na tranzistoru, jež je umístěn v portu $U_{ecp}=0,2\text{ V}$.

$U_{dd}=5\text{ V}$; $h_{fe}=420 - 800$; $I_c=100\text{ mA}$; $U_{eb}=0,7\text{ V}$; $U_{ecp}=0,2\text{ V}$; $R_{14}=?$; $P=?$

$$h_{fe} = I_c/I_b \Rightarrow I_b = I_c/h_{fe} = 100/420 = \underline{\underline{0,238\text{ mA}}}$$

$$R_{14}=(V_{dd}-U_{eb}-U_{ecp})/I_b = 4,1/0,000238 = \underline{\underline{17,23\text{ k}\Omega}}$$

Nejbližší hodnota v katalogu je **18 k Ω** .

$$P=(V_{dd}-U_{eb}-U_{ecp})\cdot I_b = 4,1\cdot 0,000238 = \underline{\underline{0,9\text{ mW}}}$$

Z toho vyplývá, že můžeme použít SMD-rezistor o velikosti 0805, který má jmenovité zatížení při 70°C 0,125 W.

5.3.2 Tlačítka na desce VDT3

Na desce VDT3 jsou umístěna tři tlačítka P-DT6SW. K tlačítkům jsou opět připojeny paralelně kondenzátory proti zákmitům kontaktů. Veškeré parametry kondenzátorů a tlačítek jsou shodné s deskou VDT2. Tlačítka na desce jsou připojena na port P3.6 až P3.4.

6. Návrh, výroba, osazení a oživení výukových desek

Plošné spoje byly navrženy pomocí návrhového systému EAGLE Layout Editor. Díky mnoha knihovnám součástek, jednoduchému a intuitivnímu ovládání programu je tento program silným nástrojem a pomocníkem při návrhu plošných spojů.

Po navržení všech tří desek byl návrh zaslán do laboratoře Technické univerzity v Liberci, kde byly desky vyleptány a prokoveny. Po vyrobení desek byly desky osazeny a zapájeny. Následně byly proměřeny všechny spoje proti zkratu a možnosti výskytu studeného spoje. Před ožívováním desky proběhla kontrola správného zapojení všech součástek. Poté každá deska prošla samostatným procesem oživení a programování.

6.1 Programování desky VDT1

Pro desku VDT1 byl navržen a odladěn následující testující program.

Na úvodní obrazovce LCD displeje se objeví nápis „VDT1 made by TUL“, který je přítomný po dobu zhruba čtyř sekund. Následně se na obrazovce LCD displeje zobrazí nápis „Stiskněte tlačítko“. Po stisku tlačítek 0 až 9 se rozsvítí příslušná dioda na L-Bargrafu odpovídající číslu na klávesnici s nápisem byla stisknuta klávesa 0 až 9. Po stisknutí klávesy „*“ začne putovat světelný „had“ z levé strany doprava se zpožděním 1 sekundy (zpoždění řešeno pomocí přerušení). Po stisku klávesy „#“ začne světelný „had“ putovat v opačném směru (had putuje zprava doleva) vždy jen jeden cyklus.

Po stisku kláves A až D se rozsvítí určité kombinace diod. Kombinace jsou následující: po stisku klávesy A se zobrazí liché diody na L-Bargrafu. Stisknutím klávesy B se rozsvítí sudé diody na L-Bargrafu. Stisknutím klávesy C se rozsvítí prostřední dvě diody a stisknutím klávesy D se rozsvítí krajní diody. Pokud dojde ke stisknutí klávesy, vždy se zobrazí na LCD displeji „Stisknuta klávesa 1 až D“

Odladěný program je umístěn na přiloženém CD nosiči ve složce programy/deska1/LCD.asm.

6.1.1 Fragmenty zdrojového kódu k desce VDT1

6.1.1.1 Zdrojový kód pro klávesnici

Funkční kód je zde uveden pouze pro dva sloupce a čtyři řádky. Stejným způsobem by se adresovaly i zbylé řádky a sloupce.

```

;-----Klávesnice-----
klav:      mov    P0,B
           mov    P3,#0ffh
           clr    P3.7           ;výběr 1.sloupce
           mov    A,P3
           anl    A,#0fh         ;smaže 7-4bit informace
           jb     Acc.3,jedna_ne  ;nestisknuto v 1.řádku
jedna_ne:  jb     Acc.2,ctyri_ne  ;vrátí 4
ctyri_ne:  jb     Acc.1,sepm_ne  ;vrátí 7
sepm_ne:   jb     Acc.0,hvezda_ne ;přeskočí na druhý sloupec
hvezda_ne: mov    P3,#0ffh
           clr    P3.6           ;vyber 2.sloupce
           mov    A,P3
           anl    A,#0fh         ;smaže 7-4bit informace
           jb     Acc.3,dva_ne    ;vrátí 2
dva_ne:    jb     Acc.2,pet_ne    ;vrátí 5
pet_ne:    jb     Acc.1,osm_ne    ;vrátí 8
osm_ne:    jb     Acc.0,nula_ne   ;vrátí 0
           ret

```

6.1.1.2 Zdrojový kód pro LCD

Ukázky použitých podprogramů pro ovládání LCD.

Kde R4 je adresace pro port P1 a R5 pro port P2

```

;-----nastavení LCD-----
main:      mov    P1,#255
           mov    P2,#255
           clr    P1.5
           call   Delay2
           ret

;-----inicializace-----
ini:       mov    R4,#00000000b   ;nastavení interface
           mov    R5,#01110011b
           call   PosliInstr

```

```

ret
;-----pošle data do LCD-----
PosliData:  setb  P1.3          ;RS na log.1 - zápis dat
            acall  PosliSpol    ;pošli displeji
            acall  Delay1       ;zpoždění 100us
            ret
;-----pošle instrukci do LCD-----
PosliInstr: clr    P1.3          ;RS na log.0 - zápis instrukce
            acall  PosliSpol    ;pošli displeji
            call   Delay2
            ret                ;vlastní vyslání bytu
;-----samotné poslání dat-----
PosliSpol:  mov    P1,R4          ;pošli data na P1
            mov    P2,R5
            nop
            setb   P1.5
            nop
            clr    P1.5
            ret
;-----Písmeno A-----
AA1:        mov    R4,#01001000b    ;A
            mov    R5,#00001011b
            call   PosliData
            ret
;-----
Ukázka adresace paměti DDRAM
;-----
ram16:      mov    R4,#10000000b    ;DDRAM na 1 řádek 6 pozice
            mov    R5,#10000111b
            call   PosliInstr
            ret

```

6.2 Programování desky VDT2

Pro desku VDT2 byly zvoleny dva demonstrační programy. První z programů po spuštění zobrazuje na segmentových displejích hodiny - v řádu hodiny, minuty a sekundy, přičemž daná data program čte z obvodu RTC po sběrnici SPI. Jako druhý demonstrační program byly zvoleny stopky. Stisknutím prvního tlačítka se stopky spustí. Po stisknutí druhého tlačítka se stopky zastaví a mohou být opět sepnuty prvním tlačítkem nebo třetím tlačítkem vynulovány. Pokud budou stopky za běhu vynulovány jejich vynulování proběhne a čítání pokračuje od nuly. Při vynulování stopky třetím tlačítkem reproduktor pípnutím oznámí vynulování. V programu stopky je jako zdroj času použito přerušení od procesoru.

Odladěné programy jsou umístěny na přiloženém CD nosiči ve složce programy/deska2/SPI1R.asm a STOPKY.asm.

6.2.1 Fragmenty kódu softwarové komunikace SPI

Procedura zápis zapisuje na adresu, která je uložena v proměnné „adr“ data, která jsou v proměnné „dta“.

```
zapis:      setb    CE           ;úvodní bit
            setb    SDO
            mov     r7,#8
            mov     a,#adr       ;posílá adresu po bitech
nav:        rlc     a
            mov     SDI,c
            setb    SCLK
            clr     SCLK
            djnz    r7,nav
            mov     r7,#8
            mov     a,#dta       ;posílá data po bitech
nav1:       rlc     a
            mov     SDI,c
            setb    SCLK
            clr     SCLK
            djnz    r7,nav1
            clr     CE
```

```

clr    SDO
call   delay
ret

```

Procedura „cteni“ čte z adresy, která je uložena v proměnné „adr1“ data do proměnné „dta1“.

```

cteni:    setb    CE
          mov     r7,#8
          mov     a,#adr1
nav2:     rlc     a
          setb    SDO
          mov     SDI,c
          setb    SCLK
          clr     SCLK
          djnz    r7,nav2
          mov     r7,#8
nav3:     mov     c,SDO
          setb    SCLK
          clr     SCLK
          rlc     a
          djnz    r7,nav3
          mov     dta1,A
          clr     CE
          call    delay
          ret

```

6.3 Programování desky VDT3

Na desce VDT3 byl testován následující program. Po stisknutí prvního tlačítka dojde k roztočení krokového motorku směrem doleva a na displeji se zobrazí písmeno „L“. Po stisku druhého tlačítka dojde k zastavení motorku a zobrazení na segmentovém displeji číslice „0“. Poslední třetí tlačítko slouží k roztočení motorku doprava a rozsvícení písmena „P“ na segmentovém displeji.

Odladěný program je umístěn na přiloženém CD nosiči ve složce programy/deska3/deska3.asm.

6.3.1 Fragment kódu pro roztočení motorku vlevo.

Jeden konec cívky na motorku je vždy v logické úrovni 1 a zbylé v logické 0. Postupným přepínáním cívek dojde k roztočení motorku.

;-----motorek vlevo-----

```
levo:      setb  A1
           clr   A2
           clr   B1
           clr   B2
           call  zpozd
           clr   A1
           clr   A2
           clr   B1
           setb  B2
           call  zpozd
           clr   A1
           setb  A2
           clr   B1
           clr   B2
           call  zpozd
           clr   A1
           clr   A2
           setb  B1
           clr   B2
           call  zpozd
           jmp   levo
```

6.4 Zadání ke cvičením

6.4.1 Deska VDT1

Navrhňte a odlaďte v jazyce symbolických adres program pro ovládání desky VDT1 v následujícím znění.

Název: „Mobilní telefon“

Kde tlačítka 1 až 9 budou mít funkci jako tlačítka na mobilním telefonu. Po stisknutí tlačítka „1“ se na první pozici LCD displeje zobrazí písmeno „A“, po druhém stisku písmeno „B“. Pokud budeme chtít pokračovat na další pozici, přesuneme se na ní potvrzením znaku. Pro potvrzení znaku bude sloužit klávesa „A“, pro smazání předchozího znaku bude sloužit klávesa „B“. Po napsání textu „Jméno Příjmení“ a stisknutí klávesy „D“ dojde pomocí přerušení k postupnému rozsvěcení součástky L-Bargaf (zleva doprava) a zobrazení textu na LCD displeji „odesílám zprávu“. Po dokončení L-Bargafu se na LCD zobrazí text „Zpráva odeslána“. Pokračovat „*” ano „#” ne. Návod na cvičení je umístěn v příloze č. 2 tohoto dokumentu.

6.4.2 Deska VDT2

Navrhňte a odlaďte v jazyce symbolických adres program pro ovládání desky VDT2 v následujícím znění.

Název: „Detonátor“

Pomocí přerušení sestavte program, který bude sloužit jako odpočet určitého časového intervalu. Posledních deset sekund začne reproduktor pípat. Po vypršení času se ozve dlouhý tón. Prvním tlačítkem se čas na detonátoru spustí. Druhým tlačítkem zastaví a po stisku třetího se detonátor nastaví na posledních 10 sekund a začne se zvukovým odpočtem. Návod na cvičení je umístěn v příloze číslo 3 tohoto dokumentu.

6.4.3 Deska VDT3

Navrhňte a odlaďte v jazyce symbolických adres program pro ovládání desky VDT3 v následujícím znění.

Název: „Automobil“

Po prvním stisknutí tlačítka 1 se roztočí krokový motorek doleva. S každým dalším jeho stisknutím se bude točení motorku zrychlovat a objeví se na displeji písmeno „P“. Po stisknutí druhého tlačítka a s jeho každým dalším stisknutím se bude motorek zpomalovat – objeví se nápis „B“. Po stisknutí třetího tlačítka dojde k zastavení motorku s nápisem „S“. Návod na cvičení je umístěn v příloze číslo 4 tohoto dokumentu.

Závěr

Tato bakalářská práce vznikla z potřeby doplnit laboratoře mikroprocesorové techniky o přípravky, na nichž si studenti vyzkouší programování různých součástek, se kterými se mohou setkat ve své praxi. Práce se zabývá popisem konstrukce prototypových desek, které byly sestaveny na základě požadavků vyučujících a konzultací s nimi. Na základě těchto požadavků byly navrženy a vytvořeny tři výukové desky připojitelné k řídicí desce ED2. Jednotlivé desky umožňují studentům vyzkoušet si velké množství jednoduchých i složitějších zadání. Každá deska je použitelná jiným způsobem. Pro otestování funkčnosti desek byly vytvořeny ukázkové programy.

Součástí této práce jsou rovněž vzorová zadání a návody pro možnost výuky na prototypových deskách. Všechny prototypy desek jsou plně funkční a splňují všechny požadavky na správnou funkčnost a bezpečnost.

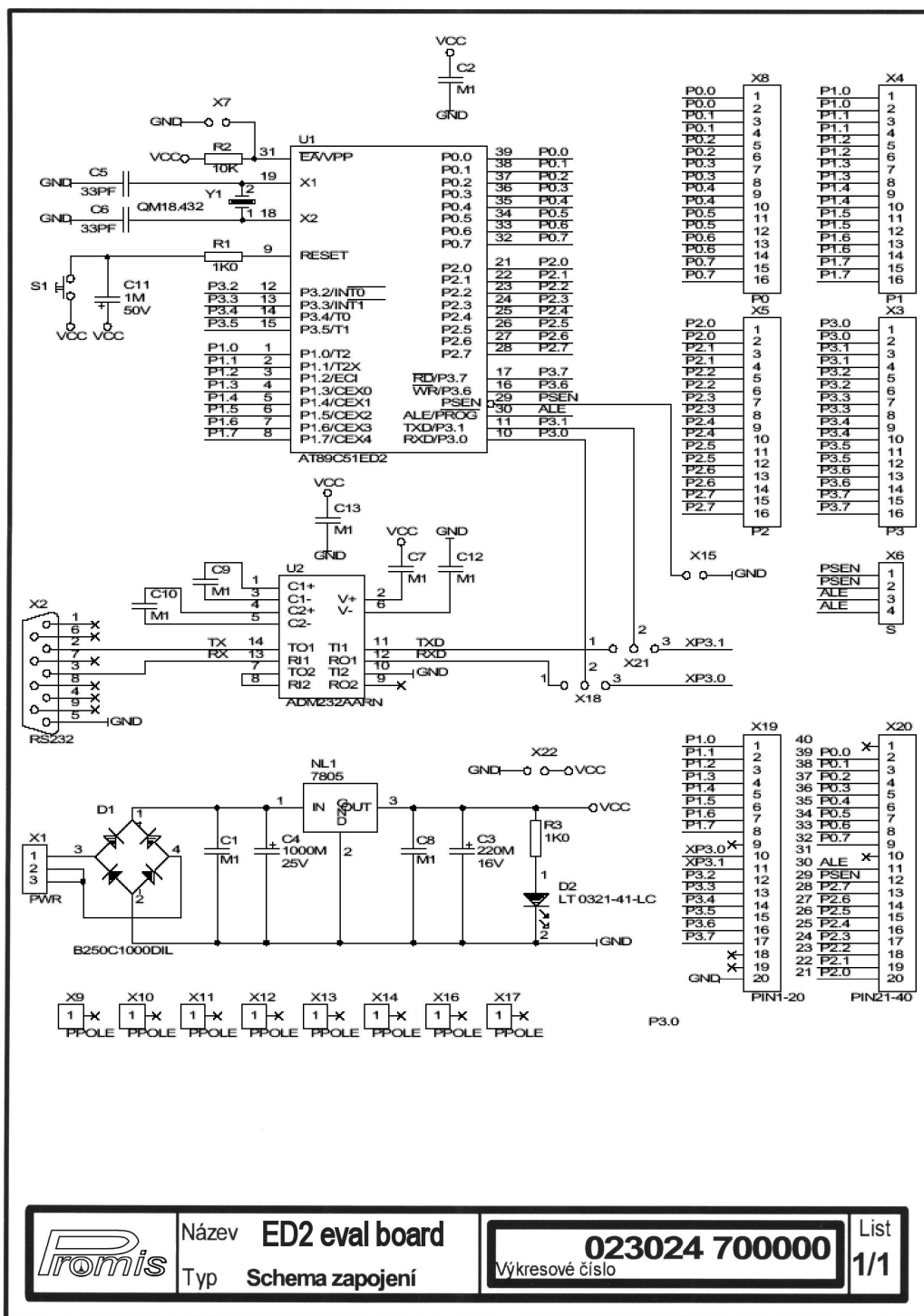
Použitá Literatura

- [1] Promis *ED2-KIT*
[online]. [cit 1.4.2007]. URL <<http://www.promislbc.cz/docs/ed2.pdf>>
- [2] Atmel Corporation, *Datasheet AT89C51RD/ED2* [online]. [cit 1.4.2007]. URL <http://www.atmel.com/dyn/resources/prod_documents/doc4235.pdf>
- [3] ČVUT, *Mikroprocesory z řady 8051*
[online]. [cit 1.4.2007]. URL <<http://noel.feld.cvut.cz/vyu/scs/prezentace99/8051/>>
- [4] Hankovec, D. *Časování CPU, obvod oscilátoru a nulování mikroprocesoru*.
[online]. [cit 1.4.2007]. URL <<http://www.dhservis.cz/casovani.htm>>
- [5] Hankovec, D. *Organizace paměti*.
[online]. [cit 1.4.2007]. URL <http://www.dhservis.cz/organizace_pameti.htm>
- [6] Hankovec, D. *Čítače a časovače*.
[online]. [cit 1.4.2007]. URL <<http://www.dhservis.cz/citace.htm>>
- [7] Hankovec, D. *Přerušovací systém*.
[online]. [cit 1.4.2007]. URL <<http://www.dhservis.cz/preruseneni.htm>>
- [8] Hankovec, D. *Vstupně výstupní porty*.
[online]. [cit 1.4.2007]. URL <<http://www.dhservis.cz/vstvyst.htm>>
- [9] Hankovec, D. *Připojení LCD displeje k 8051*.
[online]. [cit 1.4.2007]. URL <<http://www.dhservis.cz/lcd.htm>>
- [10] GM Electronic. *Součástky pro elektrotechniku 2005*
- [11] Wikipedia otevřená encyklopedie. *Reproduktor*.
[online]. [cit 1.4.2007].
URL <http://cs.wikipedia.org/wiki/Reproduktor#P.C5.99.C3.ADmovyza.C5.99uj.C3.ADc.C3.AD_-_elektrodynamick.C3.A9>
- [12] Dallas Semiconductor. *Datasheet DS1305*
[online]. [cit 1.4.2007]. URL <<http://datasheets.maxim-ic.com/en/ds/DS1305.pdf>>
- [13] Dudáček K. *Sériová rozhraní SPI, Microwire, I2C a CAN*
[online]. [cit 1.4.2007].
URL <http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf>
- [14] Hlava J. *Regulační vlastnosti elektrických pohonů a výkonových členů*
[online]. [cit 1.4.2007]. URL <<http://www.fm.tul.cz/~krsub/fm/par/pohony.pdf>>
- [15] Texas Instrument *Datasheet L293D*
[online]. [cit 1.4.2007]. URL <<http://focus.ti.com/lit/ds/symlink/l293d.pdf>>

Přílohy

1. Schéma desky ED2
2. Elektrotechnická dokumentace k VDT1; návod ke cvičení
3. Elektrotechnická dokumentace k VDT2; úprava elektrotechnické dokumentace; návod ke cvičení
4. Elektrotechnická dokumentace k VDT3; úprava elektrotechnické dokumentace; návod ke cvičení
5. Návod ke cvičení programu uScope a FLIP

Příloha č. 1
Schéma desky ED2



Příloha č. 2

Elektrotechnická dokumentace k VDT1

návod ke cvičení

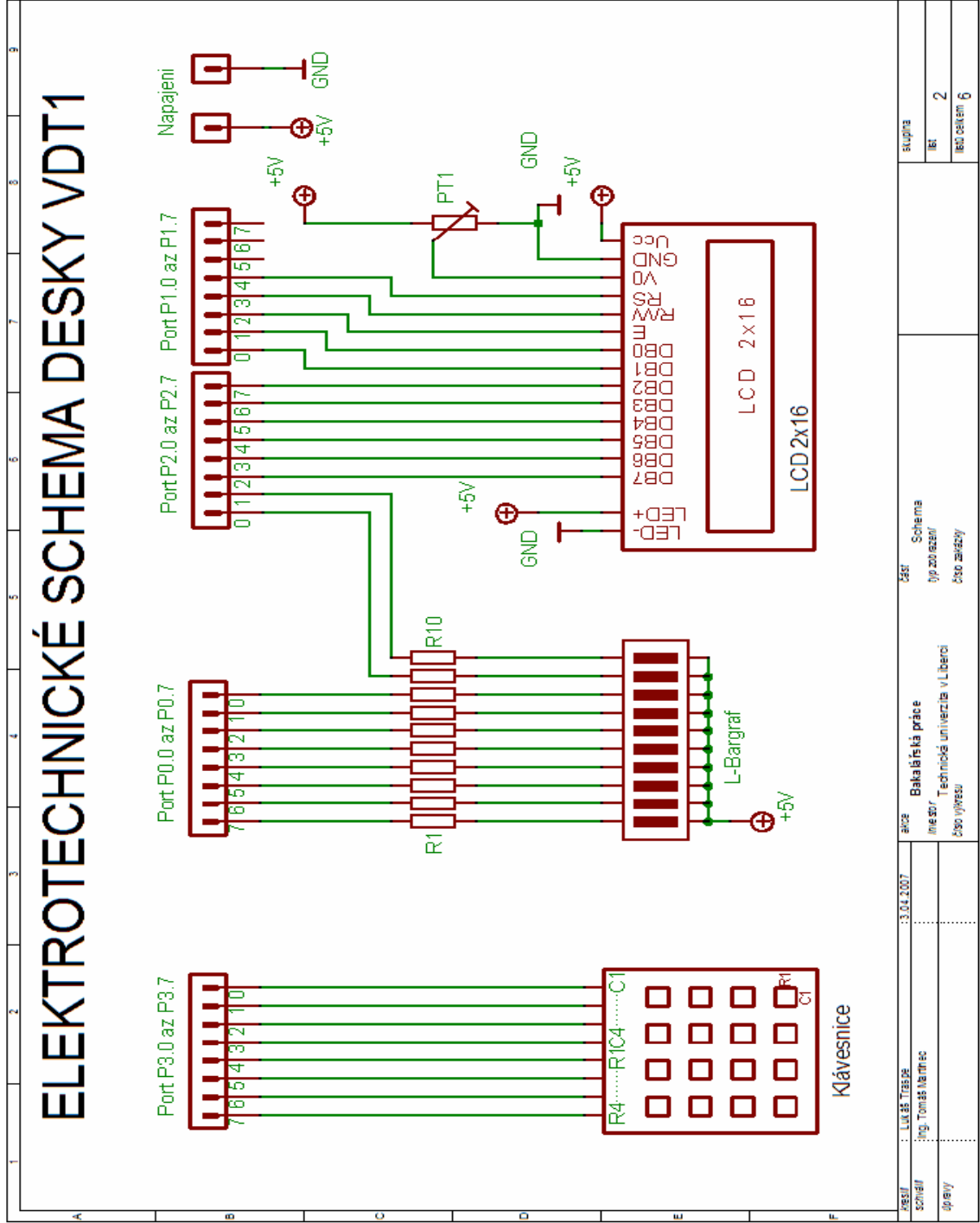
1	2	3	4	5	6	7	8	9
<h1>Bakalářská práce</h1> <h2>Elektrotechnická dokumentace k výukové desce</h2> <h3>VDT1</h3> <h2>Technická univerzita v Liberci</h2> <p>navrhl: Lukáš Traspe</p>								
Proslul : LUKÁŠ TRASPE Schválil : Ing. Tomáš Hájmanec			Datum : 13.04.2007 Miesto : Technická univerzita v Liberci Číslo výpisu :			Číslo : Schéma : Typ pozice : Číslo závazky :		
Úpravy :						Skupina : IRT : 1 IRT celkem : 6		

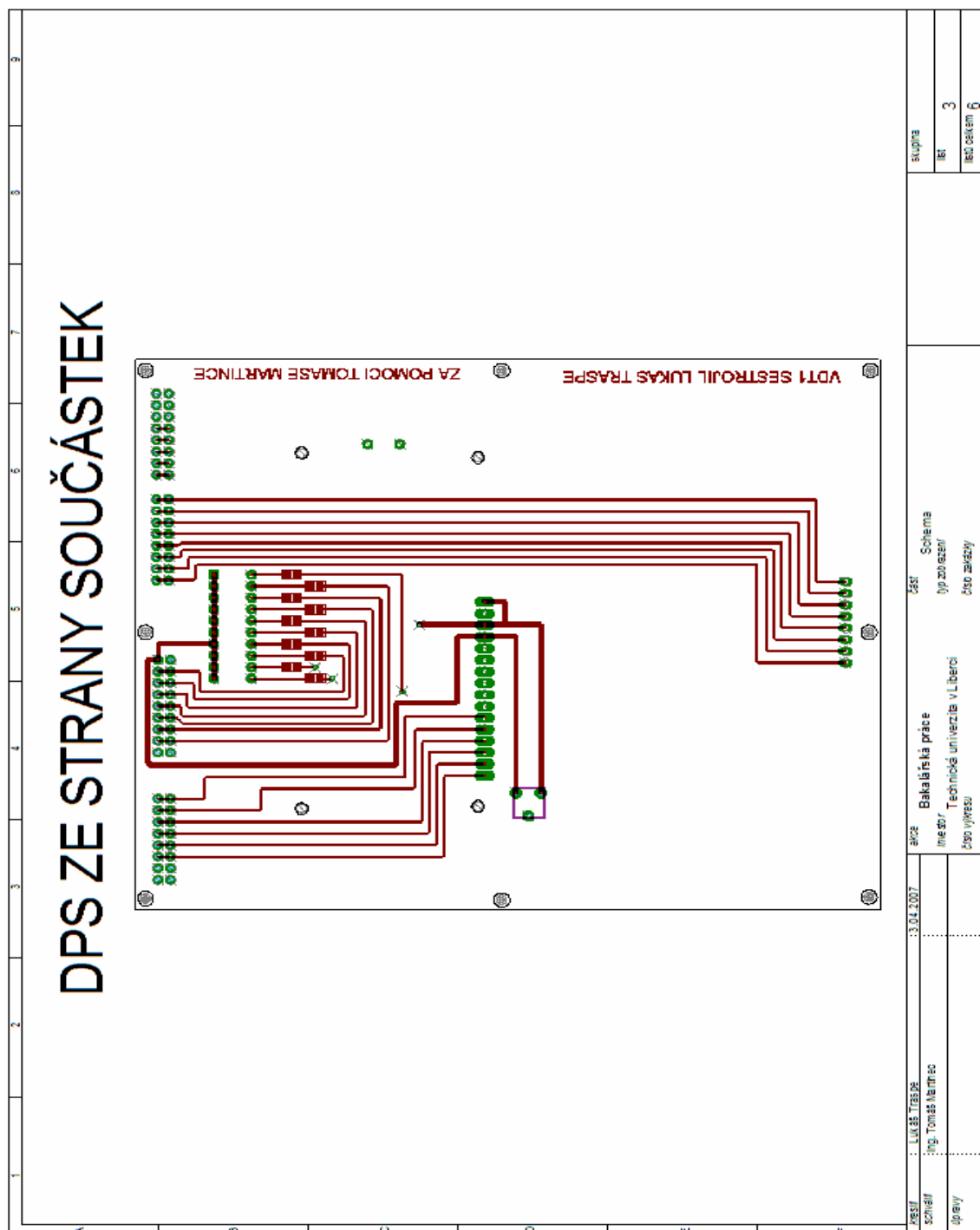
1	2	3	4	5	6	7	8	9
<h1>Bakalářská práce</h1> <h2>Elektrotechnická dokumentace k výukové desce</h2> <h3>VDT1</h3> <h2>Technická univerzita v Liberci</h2> <p>navrhl: Lukáš Traspe</p>								
Proslul : LUKÁŠ TRASPE Schválil : Ing. Tomáš Hájmanec			Datum : 13.04.2007 Miesto : Technická univerzita v Liberci Číslo výpisu :			Číslo : Schéma : Typ pozice : Číslo závazky :		
Úpravy :						Skupina : IRT : 1 IRT celkem : 6		

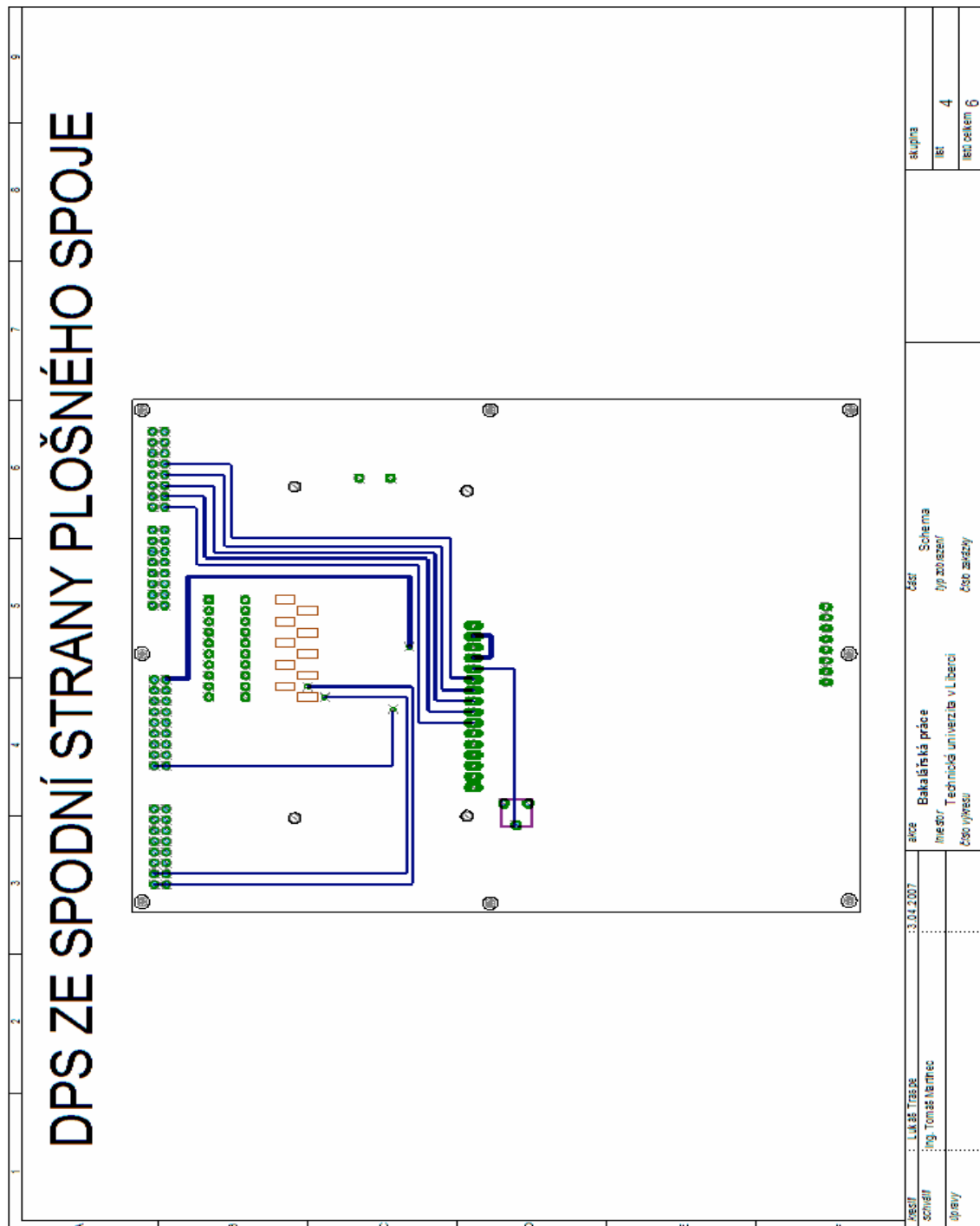
1	2	3	4	5	6	7	8	9
<h1>Bakalářská práce</h1> <h2>Elektrotechnická dokumentace k výukové desce</h2> <h3>VDT1</h3> <h2>Technická univerzita v Liberci</h2> <p>navrhl: Lukáš Traspe</p>								
Projevit : LUKÁŠ TRASPE Schválil : Ing. Tomáš Náměček			Datum : 13.04.2007			Autor : Lukáš Traspe Název : Elektrotechnická dokumentace k výukové desce VDT1 Číslo výřezu :		
Úpravy :			Technická univerzita v Liberci			Schéma : Typ rozložení : Číslo závazky :		
						Skupina : Let : 1 Počet stran : 6		

1	2	3	4	5	6	7	8	9
<h1>Bakalářská práce</h1> <h2>Elektrotechnická dokumentace k výukové desce</h2> <h3>VDT1</h3> <h2>Technická univerzita v Liberci</h2> <p>navrhl: Lukáš Traspe</p>								
Projevit : LUKÁŠ TRASPE Schválil : Ing. Tomáš Náměček			Datum : 13.04.2007			Autor : Lukáš Traspe Název : Elektrotechnická dokumentace k výukové desce VDT1 Číslo výřezu :		
Úpravy :			Technická univerzita v Liberci			Schéma : Typ rozložení : Číslo závazky :		
						Skupina : Let : 1 Počet stran : 6		

posl	LUKAS TIGER	13.04.2007	aviz	Bakalářská práce	čas	skupina
schv	Ing. Tomáš Hájek		inzer	Technická univerzita v Liberci	typ rozřazení	1
opisy			čas vyřezu	čas závěry		6







SEZNAM POUŽITÝCH SOUČÁSTEK

Označení ve schématu	Název součástky	Objednáací číslo	Počet kusů	Slovní popis
Porty	S2G20	832-023	3	Lámací lišta
R1 až R10	RR-330R SMD	901-083	10	SMD Odpor 0805
PT1	pt6vk010	112-190	1	Odporový trimr ležatý 10K 6,3mm 2,5x5mm
L-Bagraf	L-Bagraf G	512-104	1	L-Bagraf zelený
Klávesnice	F-KV-16key	637-091	1	Maticová klávesnice 24V 20mA
LCD 2x16	MC1602E-SYL	513-070	1	LCD displej Hitachi s řadičem HD44780
	KD16MX3X05	623-083	12	Dystanční sloupky M3 polyamid se závitěm

Kreslí: schválí: opravy:	Lukáš Trnava Ing. Tomáš Němec	13.04.2007	arce inženýr číslo výkresu	Bakalářská práce Tecnická univerzita v Liberci	část typ označení číslo verze	skupina	
						III	6
						listů celkem	6

Návod na cvičení na desce VDT1

1. Popis desky VDT1

Na desce VDT1 se nachází tyto programovatelné součástky.

1.L-Bargraf

2.LCD-Displej

3.Maticová klávesnice.



Obr.č.1: Deska VDT1

2. Připojení součástek na desce VDT1

L.Bargaf - je připojen na porty P0 a P2. Bráno následovně. Na první řádce je pozice diody z leva, a na druhém řádce je adresa diody na portu.

1	2	3	4	5	6	7	8	9	10
P2.1	P2.0	P0.0	P0.1	P0.2	P0.3	P0.4	P0.5	P0.6	P0.7

Tabulka č. 1: Připojení L-Bargrafu

LCD-Displej - je připojen na porty P1 a P2 a to následovně. Na prvním řádce je daný datový bit LCD-displeje a na druhém řádce je adresa pinu portu.

RS	R/W	E	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
P1.3	P1.4	P1.5	P2.2	P2.3	P2.4	P2.5	P2.6	P2.7	P1.7	P1.6

Tabulka č. 2: Připojení LCD-Displeje

Maticová klávesnice - je připojena na port P3 a to následovně. Na prvním řádce jsou uvedeny pozice maticové klávesnice kde Ř značí řádek S značí sloupec a na druhém řádce je adresa pinu portu. Sloupcem jedna je myšleno (1,4,7,*) a řádkem jedna (1,2,3,A)

Ř4	Ř3	Ř2	Ř1	S4	S3	S2	S1
P3.0	P3.1	P3.2	P3.3	P3.4	P3.5	P3.6	P3.7

Tabulka č.3: Připojení maticové klávesnice

3. Programování prvků

L-Bargaf – Tímto zdrojovým kódem bude rozsvícena dioda na první pozici součástky L-Bargraf. Dioda se rozsvítí po přivedení logické úrovně 0 na příslušný pin portu.

Zdrojový kód:

```
Org    0h
Clr    P2.1                ; rozsvítí první diodu z leva
end
```

LCD-Displej - Z dané tabulky zapojení LCD je patrné, že datové bity DB1 až DB7 nejsou na stejném portu. Proto je nutné si uvědomit jak se mají data na port posílat. Při zasílání dat nebo instrukcí na displej, musí být každá instrukce i data určitý čas na displeji, aby se požadovaná instrukce provedla. Funkce jednotlivých bitů LCD-Displeje jsou zobrazeny v následující tabulce.

RS	V log „0“ zasíláme instrukci v log „1“ zasíláme data
R/W	V log „0“ zapisujeme data v log. „1“ čteme data
E	Potvrzují bit zaslání dat nejdříve do 1 po sléze do 0
DB0 až DB7	Datové bity

Tabulka č.4: Bity LCD-Displeje

Na displej se dají posílat instrukce a data. Rozdíl mezi daty a instrukcemi je následující. Instrukce nastavuje chod a konfiguraci displeje. To znamená, že jsou nastaveny: pohyb kurzoru, smazání displeje, pozice znaku na displeji atd. Data jsou jednotlivé znaky.

Příkaz	Kód										Vysvětlivky	Délka vykonání instrukce
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Smaže displej	0	0	0	0	0	0	0	0	0	1	Smaže displej a nastaví kurzor na pozici 0.	1.64mS
Nastaví kurzor na začátek	0	0	0	0	0	0	0	0	1	*	Nastaví kurzor na pozici 0 a vynuluje posun displeje (DDRAM beze změny)	1.64mS
Nastaví vstupní režim	0	0	0	0	0	0	0	1	I/D	S	Určí směr pohybu kurzoru (I/D) a posun displeje (S). Tyto operace se provádějí během čtení/zápisu.	40uS
Zapne/vypne displej, kurzor a jeho blikání	0	0	0	0	0	0	1	D	C	B	Zapíná/vypíná displej (D), kurzor (C) a jeho blikání (B).	40uS
Nastaví pohyb kurzoru/displeje	0	0	0	0	0	1	S/C	R/L	*	*	Nastaví pohyb kurzoru nebo displeje (S/C) a směr pohybu (R/L). Obsah DDRAM zůstane beze změny.	40uS
Nastavení interface	0	0	0	0	1	DL	N	F	*	*	Nastaví délku interface (DL), počet řádků displeje (N) a znakový font (F).	40uS

Nastaví pozici v CGRAM	0	0	0	1	Adresa v CGRAM	Po tomto příkazu jsou data ze vstupu zaznamenávána do CGRAM namísto DDRAM.	40uS
Nastaví pozici v DDRAM	0	0	1		Adresa v DDRAM	Po tomto příkazu jsou data ze vstupu zapisována do a čtena z DDRAM.	40uS
Čte příznak BUSY a hodnotu adresového čítače	0	1	BF		DDRAM address	Čte příznak BUSY (BF) indikující, že displej ještě provádí některou operaci, a pozici ukazatele adresy .	0uS
Zapiše do DDRAM nebo CGRAM.	1	0			Data	Zapiše data ze vstupu DDRAM nebo do CGRAM.	40uS
Čte data z DDRAM nebo z CGRAM.	1	1			Data	Čte data z aktuální adresy DDRAM nebo CGRAM.	40uS

Tabulka č.5: Předdefinované instrukce LCD-Displeje

Zasláním instrukcí z tabulky č. 5 na porty bude provedeno nastavení LCD-Displeje. V tabulce je možné místo písmen provést volbu uživatele. Možnosti volby jsou uvedeny v tabulce č. 6.

Jméno bitu	Nastavení	
I/D	0 = Snížení pozice kurzoru	1 = Zvýšení pozice kurzoru
S	0 = displ. nepohybovat	1 = displ. Pohybovat
D	0 = Display vypnut	1 = Display zapnut
C	0 = kurzor vypnut	1 = kurzor zapnut
B	0 = blikání kurzoru vypnuto	1 = blikání kurzoru zapnuto
S/C	0 = pohybovat kurzorem	1 = pohybovat displejem
R/L	0 = rotovat vlevo	1 = rotovat vpravo
DL	0 = 4-bit interface	1 = 8-bit interface
N	0 = 1 řádkový displej	1 = 2 řádkový displej
F	0 = 5x7 bodů	1 = 5x10 bodů
BF	0 = může být zaslána instrukce	1 = Displej není připraven, provádí se vnitřní operace.

Tabulka č.6: Možnosti volby nastavení displeje uživatelem

Na LCD-Displeji je možné každý znak zapsat na určitou pozici LCD-Displeje. K tomu slouží paměť DDRAM. Na příslušnou adresu, na kterou jsou zaslána data, na té se data zobrazí. Displej má možnost zasílání až šestnácti znaků na řádek, ačkoliv jich umí zobrazit pouze osm. Bude-li na první pozici displeje zapisováno písmeno, musí být na příslušnou adresu v DDRAM zaslána datová kombinace písmene. Adresa je (00h)

DDRAM

Adresa D: 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

Adresa Ř1: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

Adresa Ř2: 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F

Znaková sada LCD Displeje

B7.....B0	0	0	0	0	0	0	1	1	1	1	1	1
xxxx0000			0	@	P	`	P	-	9	3	α	ρ
xxxx0001		!	1	A	Q	a	q	.	7	7	4	ä
xxxx0010		"	2	B	R	b	r	'	ι	7	z	β
xxxx0011		#	3	C	S	c	s]	υ	7	ε	ε
xxxx0100		\$	4	D	T	d	t	\	ε	7	7	μ
xxxx0101		%	5	E	U	e	u	.	σ	7	1	ε
xxxx0110		&	6	F	V	f	v	7	7	7	7	Σ
xxxx0111		'	7	G	W	g	w	7	7	7	7	π
xxxx1000		(8	H	X	h	x	ι	7	7	7	7
xxxx1001)	9	I	Y	i	y	7	7	7	7	7
xxxx1010		*	:	J	Z	j	z	ε	7	7	7	7
xxxx1011		+	;	K	[k	[σ	7	7	7	7
xxxx1100		,	<	L	¥	1	1	7	7	7	7	7
xxxx1101		-	=	M]	m]	7	7	7	7	7
xxxx1110		.	>	N	^	n	7	7	7	7	7	7
xxxx1111		/	?	O	_	o	7	7	7	7	7	7

Obr.č.2: Znaková sada pro LCD-Displej

Znaky, které je možné zobrazovat na LCD-Displeji se nacházejí v obrázku 2. Pro zobrazení znaku zašleme bitovou kombinaci na příslušné datové bity.

Ukázka programování LCD-Displeje: Nejdříve je nastaven displej. Následně je na šestou pozici prvního řádku zobrazeno písmeno A.

Zdrojový kód:

;------Hlavni program-----

Org	0h	
call	main	;příprava displeje
call	ini	;inicializace displeje
call	del	;vymazání displeje
call	setc	;nastavení kurzoru na pozici

```

        call    blc                ;zapne displej a kurzor blikání
        call    uvod               ;pošle na DDRAM 00h písmeno A
        jmp     konec

;-----Příprava displeje-----
main:  mov     P1,#255             ;kombinace bitů
        mov     P2,#255             ;kombinace bitů
        clr     P1.5               ;bit E
        call    Dela2              ;volá zpoždění
        ret

;-----;inicializace-----
ini:   mov     R4,#00000000b       ;nastavení interface data na P1
        mov     R5,#01110011b     ;nastavení interface data na P2
        call    PosliInstr         ;podprogram, který vyšle data z R4 a R5 na
                                   ;porty
        ret

;-----vymazání displeje-----
del:   mov     R4,#10000000b       ;smazání displeje data na P1
        mov     R5,#00000011b     ;smazání displeje data na P2
        call    PosliInstr         ;podprogram, který vyšle data z R4a R5 na
                                   ;porty
        ret

;----nastavení kurzoru na pozici----
setc:  mov     R4,#01000000b       ;nastavení kurzoru na první pozici
        mov     R5,#00000011b     ;nastavení kurzoru na první pozici
        call    PosliInstr
        ret

;---- zapnutí displeje, kurzoru, blikání ----
blc:   mov     R4,#11000000b       ;zapnutí displeje, kurzoru, blikání na P1
        mov     R5,#11000011b     ;zapnutí displeje, kurzoru, blikání na P2
        call    PosliInstr
        ret

;-----uvod-----
ram16: mov     R4,#10000000b       ;DDRAM na 1 řádek 6 pozice

```

```

    mov    R5,#10000111b      ;DDRAM na 1 řádek 6 pozice
    call   PosliInstr
    mov    R4,#01001000b      ;písmeno A
    mov    R5,#00001011b      ;písmeno A
    call   PosliData
    ret

;-----pošle data na displej-----
PosliData:
    setb   P1.3                ;RS na log.1 zápis dat
    acall  PosliSpol           ;pošli displeji
    acall  Delay1              ;zpoždění 100us
    ret

;--podprogram pošle instrukci displeji--
PosliInstr:
    clr    P1.3                ;RS na log.0 - zápis instrukce
    acall  PosliSpol           ;pošli displeji
    call   Dela2
    ret

;-----vlastní vyslání bytu-----
PosliSpol:
    mov    P1,R4               ;pošli data na P1
    mov    P2,R5               ;pošli data na P2
    nop
    setb   P1.5                ;potvrzovací bit E v log 1
    nop
    clr    P1.5                ;potvrzovací bit E v log 0
    ret

;-----zpoždění 100us -----
Delay1:
    mov    R0,#50
    djnz   R0,$
    ret

;-----zpoždění 2ms -----
Dela2: mov    R6,#20

```

```

Dela3: acall Delay1
        djnz R6,Dela3
        ret
konec: end

```

Maticová klávesnice - Programování maticové klávesnice může být prováděno různými algoritmy. Kombinace řádků a sloupců udává pozici stisknuté klávesy. Pokud bude stisknuta klávesa objeví se na pozici řádku i sloupce na příslušných bitech logická nula. Stačí tedy vytvořit testovací smyčku pomocí které bude zjištěna stisknutá klávesa.

Zdrojový kód pro první sloupec a čtyři řádky.

```

klav:  mov    P3,#0ffh          ;nastavení portu P3 do stavu kdy jsou
                                   ;všechny bity=1
        clr    P3.7             ;výběr 1.sloupce
        mov    A,P3
        anl    A,#0fh           ;smaže 7-4bit informace
        jnb    Acc.3,jed_ne     ;nestisknuto č.1 v 1.řádku
        mov    P0,#11111110b   ;po stisknutí klávesy 1 se rozsvítí jedna
                                   ;dioda na bargrafu

jed_ne:jnb    Acc.2,cty_ne
cty_ne:jnb    Acc.1,sed_ne
sed_ne: jnb    Acc.0,hvezda_ne
        jmp    klav

```

Příloha č. 3

Elektrotechnická dokumentace k VDT2

úprava elektrotechnické dokumentace

návod ke cvičení

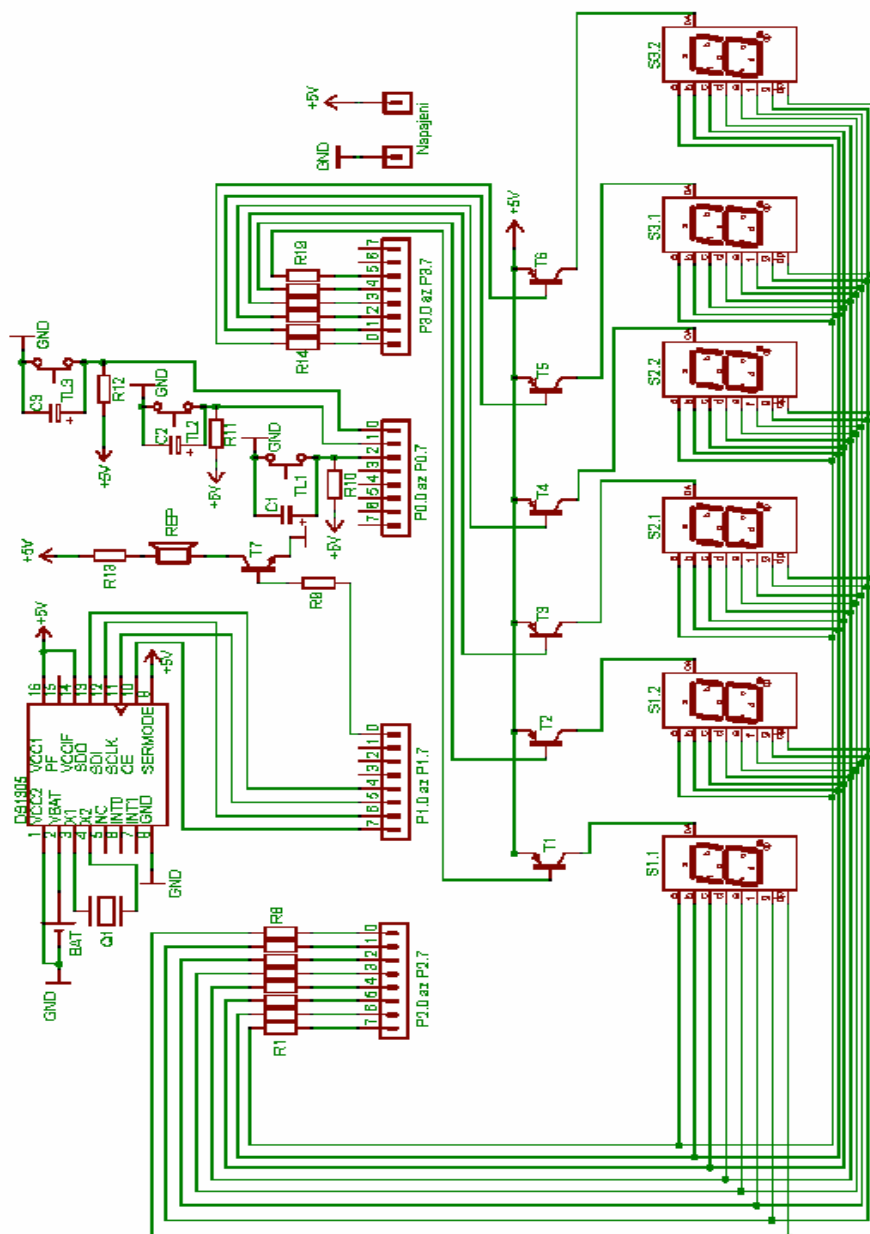
VDT2

Technická univerzita v Liberci

navrhl: Lukáš Traspe

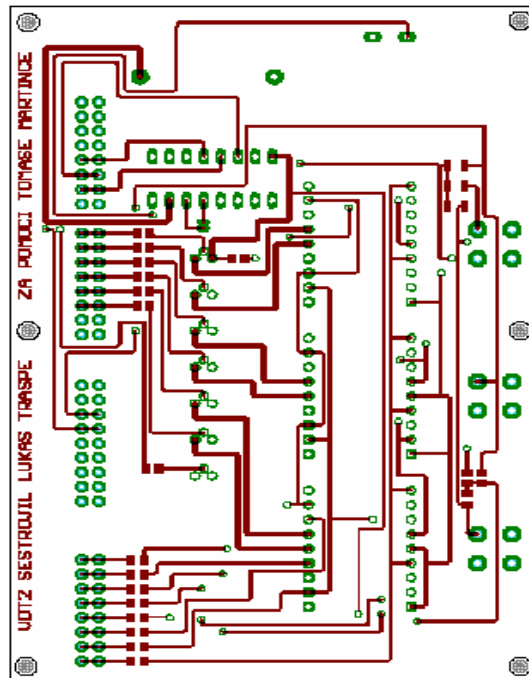
1	2	3	4	5	6	7	8	9
<h1 style="text-align: center;">Bakalářská práce</h1> <h2 style="text-align: center;">Elektrotechnická dokumentace k výukové desce</h2> <h3 style="text-align: center;">VDT2</h3> <h2 style="text-align: center;">Technická univerzita v Liberci</h2> <p style="text-align: center;">navrhl: Lukáš Traspe</p>								
KRESL : Lukáš Traspe SOVLÁD : Ing. Tomáš Martinec			AKCE : Bakalářská práce MĚSÍC : ČÍSLO VÝKRESU :		ČAS : Schéma typ obrazovky číslo stránky		skupina let počet stran	
							1 6	

ELEKTROTECHNICKÉ SCHEMA DESKY VDT2

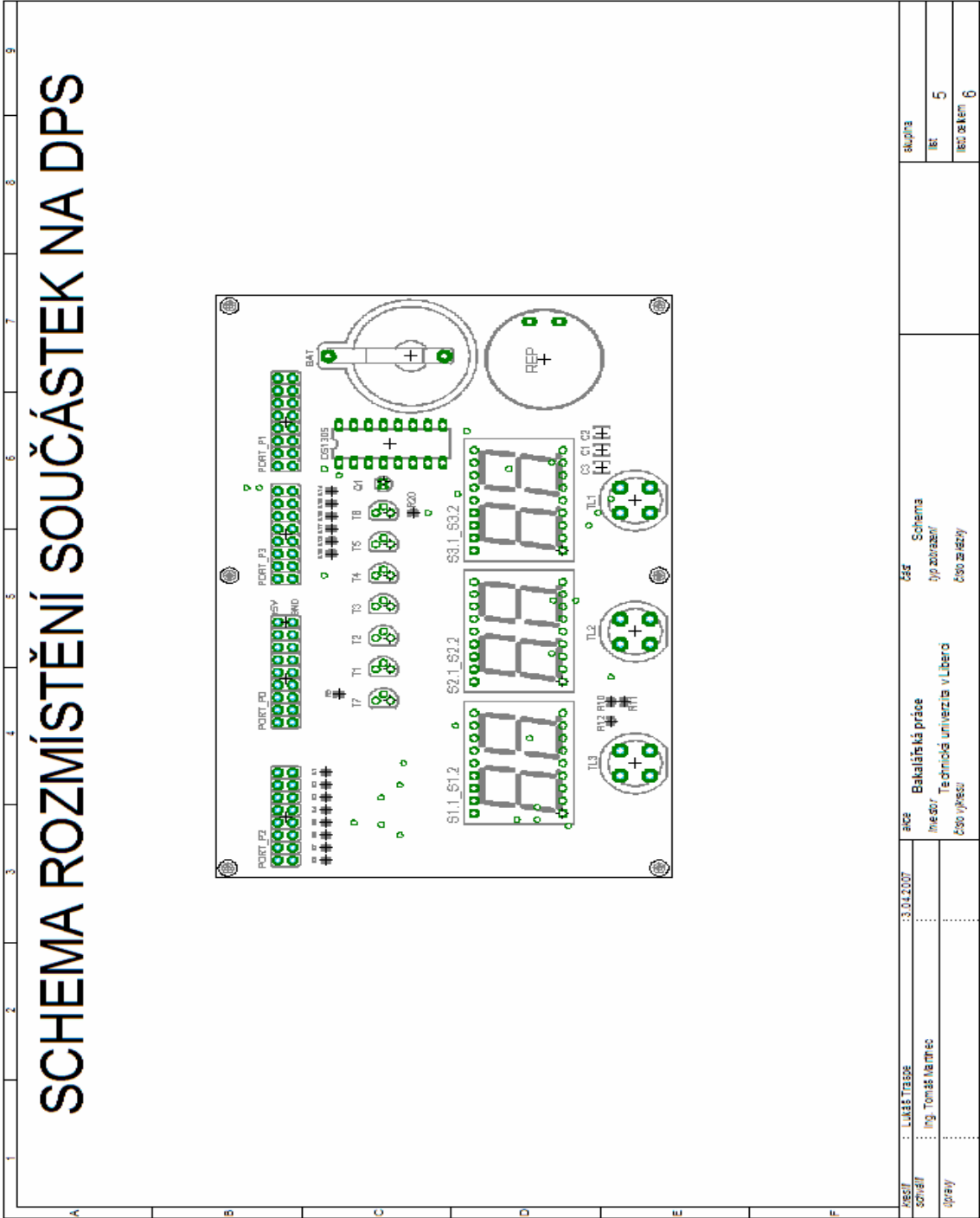


Verzi	1	Luňák Tráse	3.0.12.007	str	Bakalářská práce	str	Schema	str	6
Stránka		Ing. Tomáš Marnec		inženýr	Technická univerzita v Liberci	typ zobrazení		let	2
úpravy				číslo výkresu	číslo záznamu			inženýr	6

DPS ZE STRANY SOUČÁSTEK



Kreslí: Schválí: Opavý:	Lukáš Tráspe	13.04.2007	Práce Bakalářská práce Inženýr Ing. Tomáš Martinco Číslo výřezu	Čas Schéma Typ zobrazení Číslo zobrazení	Stručná 181 182 z 181	3 6



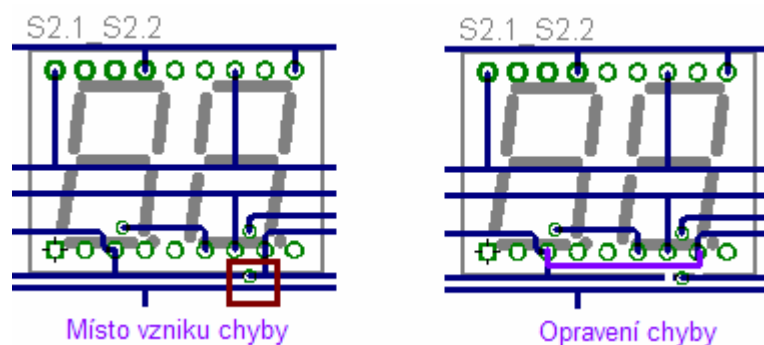
SEZNAM POUŽITÝCH SOUČÁSTEK

Označení ve schématu	Název součástky	Objednáací číslo	Počet kusů	Slovní popis
Porty	S2G20	832-023	4	Lámací lišta
R1 až R8	RR-9TR SMD	901-163	8	SMD Odpor 0805
S1.1 až S3.1	HD-A642D	512-201	3	Segmentové led displeje 14,2mm dvoumístné
T1 až T6	BC557C	210-039	6	Tranzistory u displeje PNP univerzální
T7	BC546B	210-024	1	Tranzistor u reproduktoru NPN
R14 až R19	RR-18K SMD	901-027	6	SMD Odpor 0805 u tranzistorů
R9	RR47R	110-041	1	Metalizovaný odpor 0,6W
R10 až R12	RR-4K7	901-115	3	SMD Odpor 0805 u reproduktoru
REP	LP-21008	642-051	1	Reproduktor miniaturní
C1 až C3	CTS1M/16V A	907-029	3	Kondenzátory u tlačítek SMD
TL1 až TL3	P-DT6SW	630-048	3	Tlačítka-černá kulatá
DS1305	DS1305	Maxim	1	Vzorek Maxim
Q1	Q32,768KHz	131-036	1	Krystal
BAT	BH2032	819-043	1	Bateriové pouzdro
	KDI6MX3X05	623-083	6	Dystanční sloupek M3 polyamid se závitem
	B-CR2032	542-006	1	Baterie

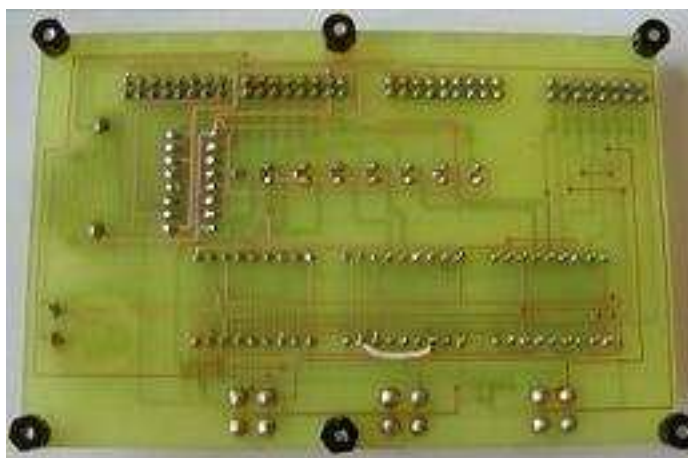
projekt	LM445 Třesce	datum	13.04.2007	strana	skupina
autor	Ing. Tomáš Hájek	inženýr	Bakalářská práce	typ zobrazení	list
opráv		číslo výkresu	Technická univerzita v Liberci	číslo strany	list ze listů
					6

Úprava elektrotechnické dokumentace desky VDT2

U desky VDT 2 byla upravena elektrotechnická dokumentace z důvodu vzniklé chyby při návrhu desky plošných spojů. Tato chyba na již vyrobené desce musela být odstraněna následujícím způsobem.



Obr. č. 1: Schéma opravy



Obr. č. 2: Oprava chyby na desce VDT2

Chyba byla odstraněna následujícím způsobem: Odstranění cestičky u osmého pinu segmentového displeje a přerušení vodivé cesty nalevo od prokoveného bodu. Třetí a osmý pin segmentového displeje byl propojen pomocí drátku.

Chyba by měla za následek rozsvícení dvou segmentů na segmentovém displeji při úmyslném sepnutí pouze jednoho ze segmentů.

Návod na cvičení na desce VDT2

1. Popis desky VDT2

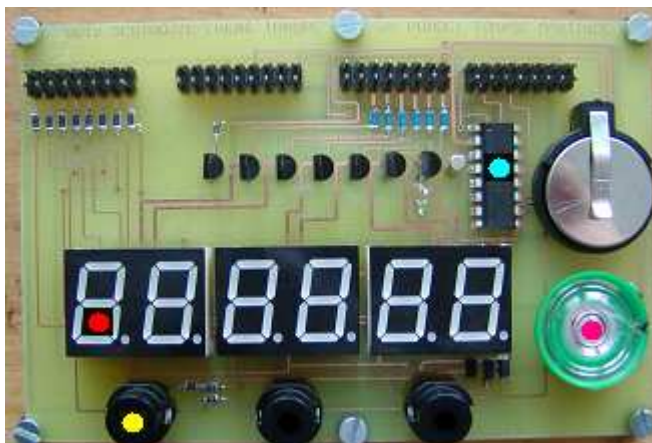
Na desce VDT2 se nachází tyto programovatelné součástky:

1. Tlačítka

2. Sedmisegmentové displeje

3. Reprodukční

4. Obvod reálného času (RTC)



Obr.č.1: Deska VDT2

2. Připojení součástek na desce VDT2

Tlačítka jsou připojena na portu P0. První řádek v tabulce udává tlačítka zleva. A druhý řádek udává bit na kterém je tlačítko připojeno.

TL1	TL2	TL3
P0.0	P0.1	P0.2

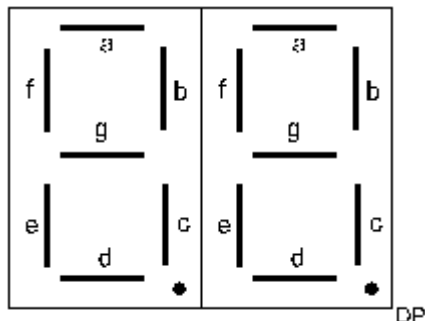
Tabulka č.1: Připojení tlačítek

Tlačítko je stisknuto pakliže se na příslušném pinu procesoru objeví logická úroveň o hodnotě 0.

Sedmisegmentové displeje - V každém segmentovém displeji se nachází osm světelných segmentů (osm diod) jejich kombinací jsou rozsvíceny potřebné znaky. Z důvodu nedostatku pinů procesoru jsou všechny světelné segmenty připojeny na jeden port, a to port P2. To znamená, že při posílání dat například písmena A na port P2 by svítily všechny segmentové displeje najednou. Aby nedocházelo k tomuto jevu, který je nežádoucí, jsou jednotlivé segmentové displeje spínány z jiného portu a to z portu P3. Pokud tedy na portu P3 dojde k sepnutí jednoho segmentového displeje budou zasílaná data svítit pouze na vybraném displeji. Při zobrazování rozdílných dat na všech displejích je nezbytné přepínat dané displeje takovou rychlostí, aby lidské oko nepostřehlo zhasnutí daného segmentu (alespoň 25x za vteřinu). Jednotlivé segmentové displeje a jejich segmenty se aktivují vždy, když je na příslušný bit poslána logická úroveň o hodnotě 0. Každá dioda v sedmisegmentovém displeji má přiřazeno vlastní

označení (písmenko v abecedě). Pokud bude požadováno zobrazení čísla „A“ musí uživatel na port P2 poslat tento kód:

a b c d e f g dp
0 0 0 1 0 0 0 1



Obr.č.2: Popis jednotlivých segmentů

Jednotlivé segmenty jsou připojeny na tyto bity na portu P2.

A	b	c	d	e	f	g	dp
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0

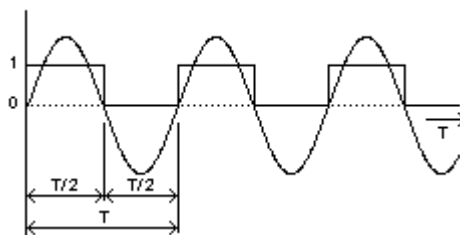
Tabulka č.2: Připojení segmentů

Bity které rozhodují o tom jaký displej bude sepnut jsou na portu P3. První řádek určuje číslo displeje zleva. A druhý určuje jeho adresu.

SEGD1	SEGD2	SEGD3	SEGD4	SEGD5	SEGD6
P3.5	P3.4	P3.3	P3.2	P3.1	P3.0

Tabulka č. 3 - Řídících bitů pro výběr displeje

Reproduktor je připojen na port P1.7. K ovládání reproduktoru z portu procesoru je použit obdélníkový signál s naprogramovanou časovou periodou. Perioda signálu určuje výšku tónu. Počet period signálu určuje délku tónu. Během jedné periody může být vyslán stejný počet jedniček i nul (logických úrovní), tím vzniká symetrický signál. Samozřejmě je možné vytvořit také asymetrický signál, který je tvořen rozdílným počtem jedniček a nul během jedné periody. Na obrázku č. 3 je zobrazen obdélníkový signál jehož jednu periodu tvoří logická úroveň jedna a druhou logická úroveň nula.



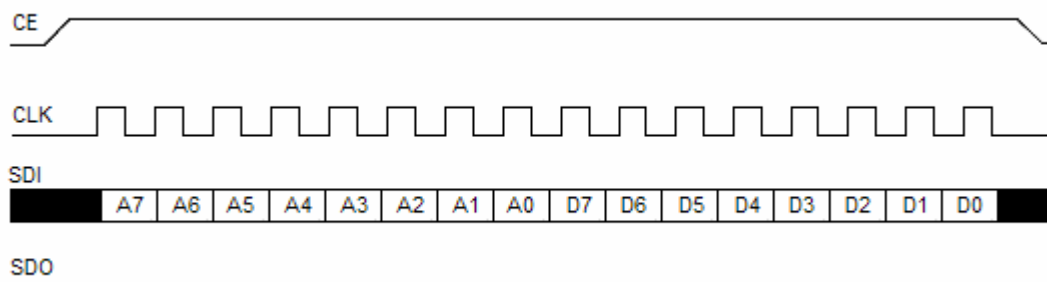
Obr.č.3: Průběh symetrického signálu

Obvod reálného času - RTC (real time clock). Jedná se o obvod, který dokáže po své aktivaci ve vnitřní struktuře uchovávat reálný čas. Obvod je připojen na portu P1. Na prvním řádku je název pinu a na druhém je jeho adresa na portu P1.

SDO	SCLK	SDI	CE
P1.4	P1.5	P1.6	P1.7

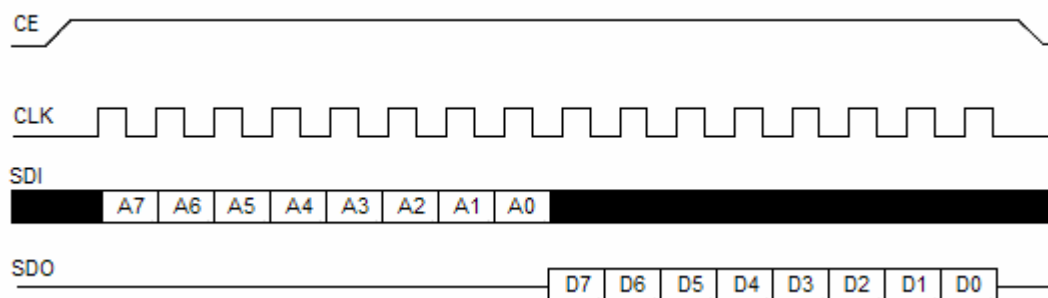
Tabulka č. 4: Ovládací signály obvodu RTC

Piny, které jsou zobrazeny v tabulce slouží pro komunikaci mezi obvodem RTC a procesorem po sběrnici SPI. Pro programování obvodu RTC byla zvolena softwarová komunikace po sběrnici SPI. To znamená, že všechna data, adresy a potvrzující bity musejí být řízeny programově.



Obr.č.4: Zápis dat do paměti obvodu RTC

Při softwarovém řízení zápisu dat do obvodu RTC po sběrnici SPI musí být nejdříve přiveden signál CE a SDO do logické úrovně 1 následně je vyslán signál CLK a zapsán daný bit SDI. Takto jsou zapsána všechna potřebná data do obvodu RTC. Po skončení posílání dat je signál CE uveden do logické úrovně 0. Podobným způsobem se i data z obvodu RTC získávají (Obr. 5).



Obr.č.5: Čtení dat z paměti obvodu RTC

Nejdříve je přiveden signál CE do logické úrovně „1“. Následně je zaslán signál CLK, poté signál SDI. Po zaslání všech bitů adresy je s hodinovým signálem zahájeno čtení příslušných dat ze signálu SDO. Po dokončení operace je nutné přivést signál CE do logické úrovně 0. V tabulce číslo 5 jsou uvedeny všechny adresy určené pro čtení a zápis u obvodu RTC.

Hexadecimální adresa		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Rozsah
READ	WRITE									
00H	80H	0	desítky sekund			sekundy			00-59	
01H	81H	0	desítky minut			minuty			00-59	
02H	82H	0	12	P	desítky hodin	hodiny			01-12 + P/A	
				A					00-23	
				24					10	
03H	83H	0	0	0	0	den			1-7	
04H	84H	0	0	desítky data		datum			1-31	
05H	85H	0	0	desítky měsíců		měsíc			01-12	
06H	86H	desítky let				rok			00-99	
-	-	alarm 0								-
07H	87H	M	desítky sekund alarmu			sekundy alarmu			00-59	
08H	88H	M	desítky minut alarmu			minuty alarmu			00-59	
09H	89H	M	12	P	desítky hodin	hodinový alarm			01-12 + P/A	
				A					00-23	
				24					10	
0AH	8AH	M	0	0	0	denní alarm			01-07	
-	-	alarm 1								-
0BH	8BH	M	desítky sekund alarmu			sekundy alarmu			00-59	
0CH	8CH	M	desítky minut alarmu			minuty alarmu			00-59	
0DH	8DH	M	12	P	desítky hodin	hodinový alarm			01-12 + P/A	
				A					00-23	
				24					10	
0EH	8EH	M	0	0	0	denní alarm			01-07	
0FH	8FH	kontrolní registr								-

10H	90H	stavový registr	-
11H	91H	nabíjecí registr	-
12-1FH	92-9FH	rezervováno	-
20-7FH	A0-FFH	96 bytů uživatelské RAM	00-FF

2. Programování prvků

Tlačítka – Program otestuje všechny tři tlačítka. Po stisku prvního zavolá podprogram zvuk.

Zdrojový kód:

```

Org    0h
klav:  jb    P0.0,prvne           ;nestisknuto v 1.řádku
        call  zvuk                ;zavolá po stisku tlačítka podprogram zvuk
prvne:  jb    P0.1,drune
drune:  jb    P0.2,tretine
tretine: jmp   klav

```

Sedmisegmentové displeje – Program zobrazí na prvním displeji písmeno 1 a na druhém displeji písmeno 0.

Zdrojový kód:

```

Org    0h
zacatek: mov  P3,#11011111b      ;aktivuje první segmentový displej
        mov  P2,#00111111b      ;zobrazí na displeji 1
        call delay
        mov  P3,#11101111b      ;aktivuje druhý segmentový displej
        mov  P2,#00000011b      ;zobrazí na displeji 0
        call delay
        jmp  zacatek
delay:  mov  R5,#10              ;zpoždění
        djnz R5,$
        ret

```

Reproduktor - Program zasílá symetrický signál na port P1.0.

Zdrojový kód:

```
                Org    0h
zvuk:  setb    P1.0                ;pošle na P1.0 log 1
                mov    R7,#100      ;posílá 100 cyklů
                djnz   R7,$
                clr    P1.0          ;pošle na P1.0 log 0
                mov    R7,#100      ;posílá 100 cyklů
                djnz   R7,$
                jmp    zvuk
```

Obvod reálného času (RTC) - Tento program ukazuje softwarovou komunikaci mezi obvodem RTC a procesorem. Čtení hodnot z obvodu RTC.

Zdrojový kód:

```
                Org    0h
CE      bit    P1.7
SDI     bit    P1.6
SCLK    bit    P1.5
SDO     bit    P1.4
cteni:  setb    CE                ;čtení adresy po sběrnici SPI
                call  delay        ;zavolá zpoždění
                mov   r7,#8        ;nastaví 8 cyklů pro 8 bitů
nav2:   rlc     a                ;rotace do leva obsahu akumulátoru
                setb  SDO          ;SDO do 1
                setb  SCLK         ;SCLK do 1
                mov   SDI,c        ;do SDI zaslán bit C z akumulátoru
                clr   SCLK         ;SCLK do 0
                djnz  r7,nav2
                mov   r7,#8
nav3:   setb    SCLK              ;čtení dat po sběrnici SPI
                mov   c,SDO        ;uloží do bitu C přečtenou hodnotu z RTC
                clr   SCLK         ;SCLK do 0
```

```
rlc    a                ;přečtená data po cyklu jsou v Acc
djnz   r7,nav3
clr    CE
jmp     cteni
```

Pro zápis může být použit stejný algoritmus s malou úpravou viz obr. č.4.

Příloha č. 4

Elektrotechnická dokumentace k VDT3

úprava elektrotechnické dokumentace

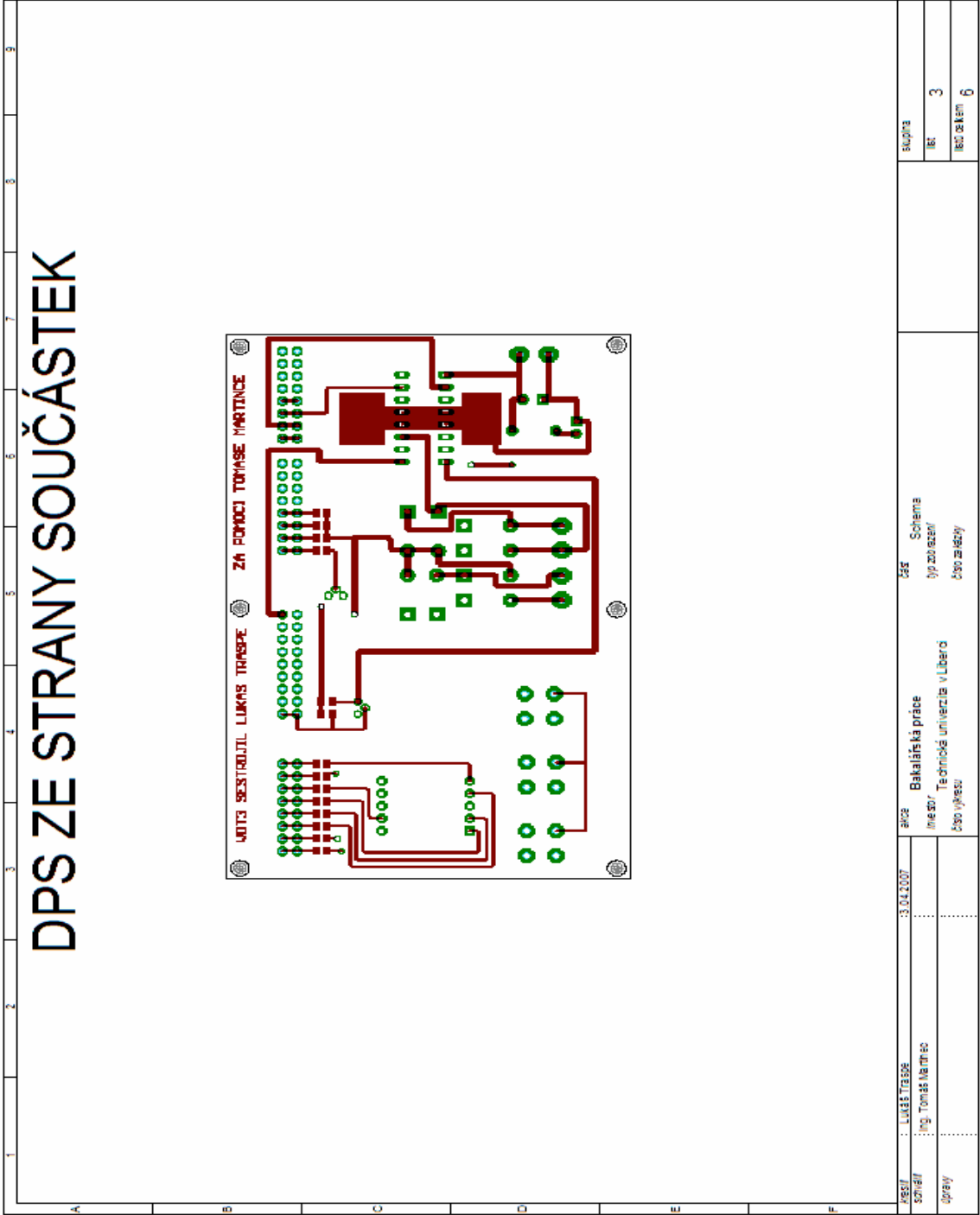
návod ke cvičení

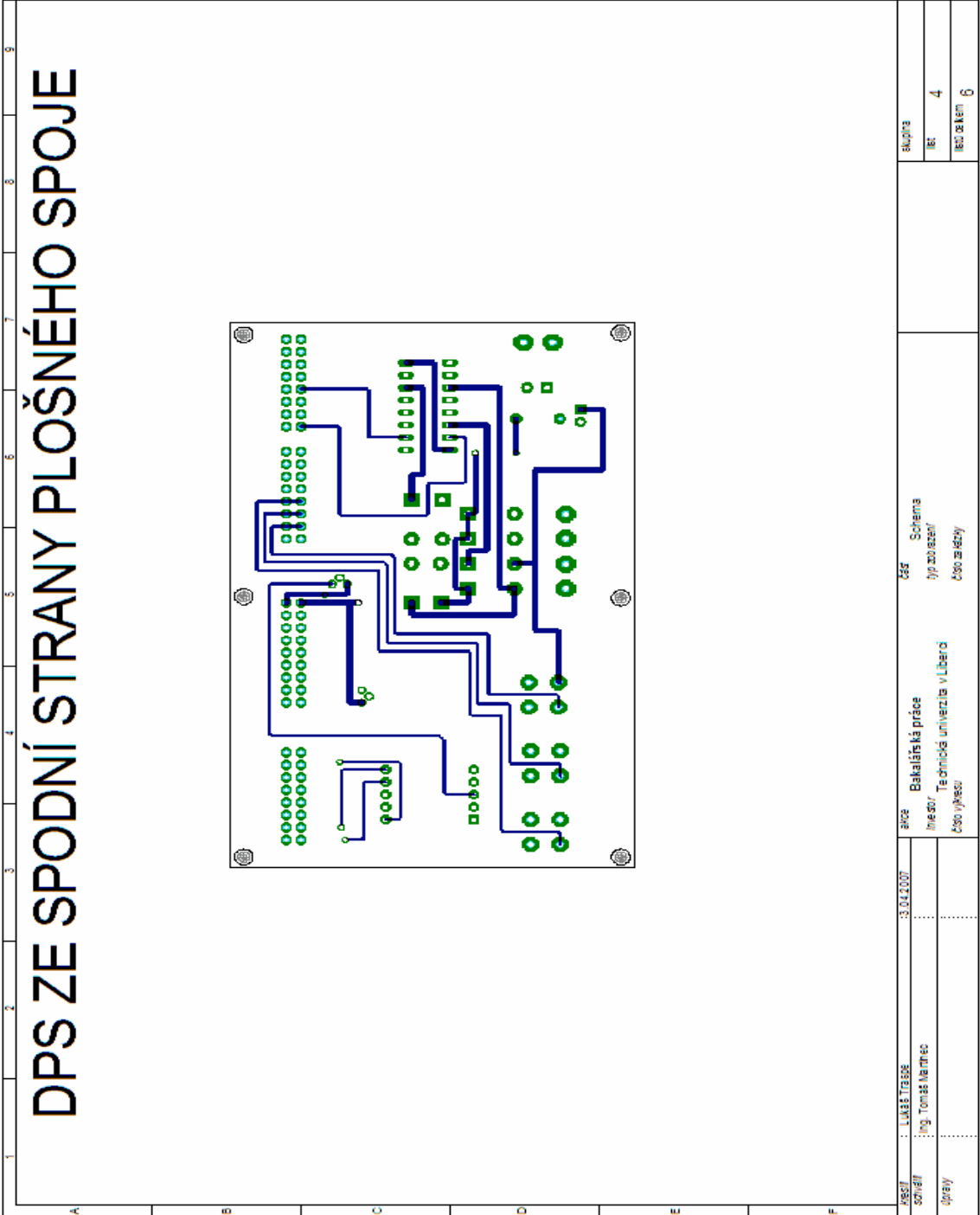
VDT3

Technická univerzita v Liberci

navrhl: Lukáš Traspe

hesl	: Lukáš Trávníček	3.04.2007	avce	Bakalářská práce	čas	Schema	strana
ovněn	: Ing. Tomáš Matrnec			hesl	typ obsahu	1	
uprav					Technická univerzita v Liberci	číslo stránky	6
					číslo strany		





SCHEMA ROZMÍSTĚNÍ SOUČÁSTEK NA DPS

SEZNAM POUŽITÝCH SOUČÁSTEK

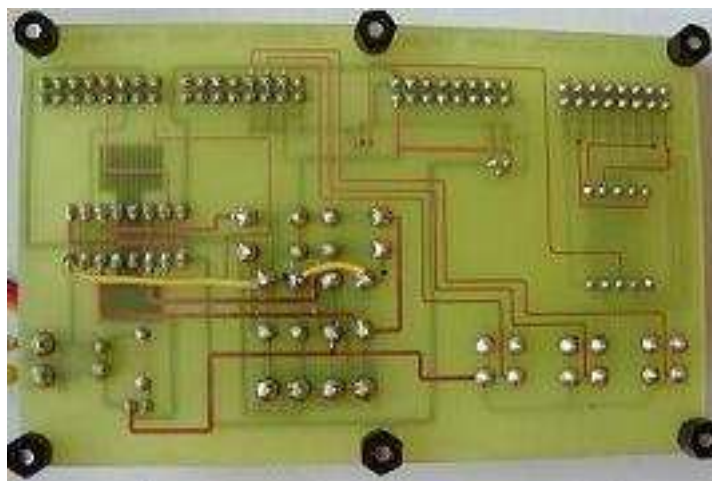
Označení ve schématu	Název součástky	Objednáací číslo	Počet kusů	Slovní popis
C1 až C3	E100M/10VM	123-220	3	Kondenzátory u tlačítek SMD
TL1 až TL3	P-DT6SW	630-048	3	Tlačítka-černa kulatá
D2 až D9	BY296	223-074	8	Rychlá dioda 1,5A 100V
R1 až R8	RR-240R SMD	901-059	8	SMD Odpor 0805
T1	BC557C	210-039	1	NF univerzální tranzistor PNP
SEG	HD-A52RD	512-181	1	LED displej 14,2mm zelený
K1 až K3	ARK500/2	821-017	3	Svorkovnice ARK mini 2 póly
	S2G20	832-023	3	Lámací lišta
	KD16MX3X05	623-083	6	Dystanční sloupek M3 polyamid se závitem
C4	E100M/50V	123-122	1	Kondenzátor elektrolitický radiální 3,5mm
D1	LED3MM2MA/G	511-200	1	Dioda signalizující napájení 24V
R13	3K3	110-075	1	Odpor k diodě
L239D	L293D	399-017	1	Čtyřkanálový můstkový budič DIL-16
T2	BC547B	210-026	1	NF univerzální tranzistor NPN
R10	RR-4K7 SMD	901-115	1	SMD Odpor 0805
R11	RR-10K SMD	901-176	1	SMD Odpor 0805
R9	RR-18K SMD	901-027	1	SMD Odpor 0805
	MW9112UGS	751-043	1	Adapter 9V/1,2A

KESU	Ing. Lukáš Trásko	3.04.2007	anot	Bakalářská práce	čas	skupina
schéma	Ing. Tomáš Hartmiec		linea sor	Technická univerzita v Liberci	typ zobrazení	let
opravy			číslo výkresu		číslo záznamu	6

Úprava elektrotechnické dokumentace desky VDT3

U desky VDT 3 byla upravena elektrotechnická dokumentace z důvodu vzniklé chyby při návrhu schématu plošného spoje. Vzniklá chyba neměla vliv na správnou funkčnost desky. Její odhalení vzniklo při podrobné kontrole zapojení.

Chyba vznikla mezi ochrannými diodami (D3, D4) a (D7, D8). Mezi tyto diody mělo být přivedeno napětí ze zdroje napájení krokového motorku. Kdyby tato chyba nebyla odhalena hrozilo by riziko zničení obvodu L293D. Pokud by se na vinutí motorku vytvořilo napětí induktivního charakteru a nedošlo by k jeho odvedení zpět do zdroje, obvod by byl s velkou pravděpodobností zničen.



Obr.č.1: Opravená chyba na desce VDT3

Tato chyba, na již vyrobené desce, byla odstraněna následujícím způsobem: jednotlivé dvojice diod byly propojeny propojkou a druhou propojkou byly připojeny na napájecí napětí přídatného zdroje.

Návod na cvičení na desce VDT3

1. Popis desky VDT3

Na desce VDT3 se nachází tyto programovatelné součástky.

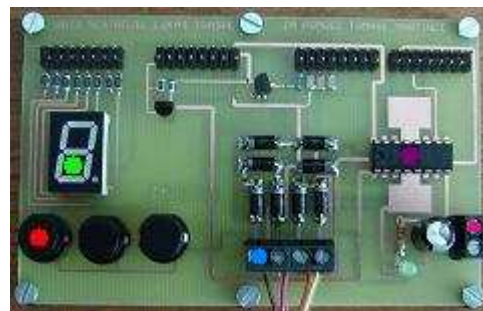
1.Tlačítka

2.Sedmisegmentový displej

3.Obvod s integrovanými H-můstky.

4.Svorkovnice pro připojení cívek motorku

5.Svorkovnice pro přivedení napájení pro motorek



Obr.č.1: Deska VDT2

2. Připojení součástek na desce VDT3

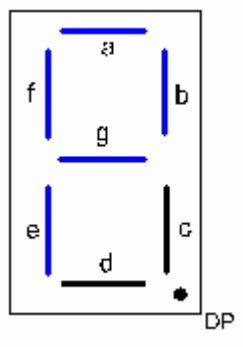
Tlačítka - jsou připojena na portu P3. První řádek v tabulce udává tlačítka zleva. A druhý řádek udává bit, na kterém je tlačítko připojeno.

TL1	TL2	TL3
P3.6	P3.5	P3.4

Tabulka č.1: Připojení tlačítek

Tlačítko je stisknuto pakliže se na příslušném pinu procesoru objeví logická úroveň o hodnotě 0

V **sedmisegmentovém displeji** se nachází osm světelných segmentů (osm diod) jejich kombinací se rozsvítí potřebné znaky. Segmentový displej je aktivní, když je na bit P3.7 přivedena logická úroveň o hodnotě 0. Jednotlivé diody jsou aktivovány rovněž přivedením logické úrovně 0 na příslušný pin portu P2



Obr.č. 2: Označení jednotlivých segmentů

Každý bit na portu P2.0 až P2.7 určuje jednu diodu na segmentovém displeji. V prvním řádku jsou diody abecedně označeny (viz. obr. 1) a ve druhém řádku jejich adresa. Ve třetím poté bitová kombinace pro rozsvícení písmena P.

a	b	c	d	e	f	g	DP
P2.0	P2.1	P2.2	P2.3	P2.4	P2.5	P2.6	P2.7
0	0	1	1	0	0	0	1

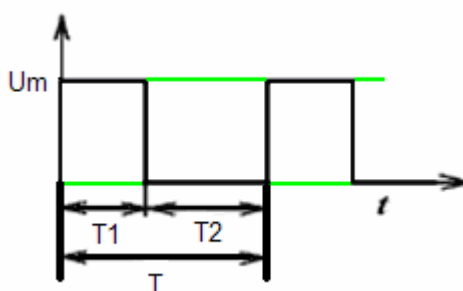
Tabulka č. 2: Připojení segmentů na port

Obvod s integrovanými H-můstky. Pomocí tohoto obvodu je možné ovládat krokové motorky. Obvod je zapojen na port P1. Na tento port jsou přivedeny piny, kterými jsou ovládány jednotlivé cívky motorku. Na portu P0.7 je pin, který spíná výkonovou část z přídatného zdroje. Na prvním řádku tabulky č. 3 jsou popsány bity pro ovládání cívek a povolení výkonové části. Na druhém řádku je jejich adresa na portu.

A1	A2	A3	A4	1-2EN a 3-4EN
P1.7	P1.6	P1.5	P1.4	P0.7

Tabulka č. 3: Signály pro ovládání krokového motorku

Ovládání motorku pomocí PWM (pulzně šířková modulace) je proces, při kterém dochází k poměrné změně délky pulzů o konstantní frekvenci. To znamená, že perioda signálu je pořád stejná, rychlost otáčení motoru se však mění změnou času T1 a T2 od minima do maxima. Pokud bude potřeba točit motorem co nejrychleji, co nejvíce zvětšíme čas T1 na úkor T2, pokud bude potřeba dosáhnout opačného jevu, bude muset být zvětšena velikost času T2 na úkor času T1.



Obr.č.3: Pulzně šířková modulace

Při vhodné rychlosti posílání pulsů se motorek plynule točí. Při vyšší rychlosti posílání pulsů se motorek nestihne roztočit a neběží plynule nebo se motorek vůbec neroztočí. Čím pomaleji se posílají pulsy, tím pomaleji se motorek točí. Při malých rychlostech se začne motorek otáčet jen o určitý krok. Napájecí napětí pro krokový motorek je 5V.

Na **svorkovnici pro připojení cívek motorku** se připojují cívky motorku. Pokud bude připojována jedna cívka motorku připojíme jeden konec na první svorku a druhý konec cívky na druhou svorku. Další cívka bude připojena na svorku 3 a 4.

POZOR při zapnutém napájení je zakázáno dotýkat se plošného spoje a svorkovnic motorku. NEBEZPEČÍ ÚRAZU ELEKTRICKÝM PROUDEM!

Svorkovnice pro přivedení napájení pro motorek. Na první označenou svorku se přivádí stejnosměrné napájecí napětí podle typu motorku. Na druhou svorku se přivede ze zdroje 0V. Jedná-li se o motorek z disketové mechaniky je přivedeno napětí o hodnotě 5V. Deska je konstruována až na hodnotu napájení 24V. Signalizaci přídavného napájení obstarává dioda umístěna za svorkami přídavného napájení.

3. Programování součástek

Tlačítka - Po stisknutí prvního tlačítka se na displeji objeví písmeno P.

Zdrojový kód:

```
Org    0h
clr    P3.7                ;aktivuje segmentový displej
klav:  mov    A,P3
        jb     Acc.6, ne     ;nestisknuta 1. klávesa
        mov    P2,#10001100b ;zobrazí písmeno P
ne:     jmp     klav
```

Sedmisegmentový displej

Na displeji se po spuštění programu objeví písmeno P.

Zdrojový kód:

```
Org    0h
st:     clr    P3.7                ;aktivuje segmentový displej
mov     P2,#10001100b             ;zobrazí písmeno P
        jmp     st
```

Obvod s integrovanými H-můstky

Po spuštění programu se motorek točí doprava a na displeji svítí písmeno P.

Zdrojový kód:

```
Org    0h
A1     bit P1.7                ;přiřazení bitů názvům
A2     bit P1.6
B1     bit P1.5
B2     bit P1.4
clr    P3.7

start:
mov    P0,#00000000b          ;povolení točení motorku
call   pravo
jmp    start

;-----motorek pravo-----

pravo:
setb   A1                    ;postupné spínání cívek u motorku
clr    A2
clr    B1
clr    B2
call   zpozd
clr    A1
clr    A2
setb   B1
clr    B2
call   zpozd
clr    A1
setb   A2
clr    B1
clr    B2
call   zpozd
clr    A1
clr    A2
clr    B1
```

```

        setb    B2
        call    zpozd
        jmp     pravo
        mov     R1,#10
zpozd:  mov     R2,#50                ;zpoždění mezi posíláním pohybových
                                       ;hodnot
zpoz:   djnz    R2,zpoz
        djnz    R1,zpozd
        ret

```


Příloha č. 5

Návod ke cvičení k programu uScope a FLIP

Postup práce s programem uScope a FLIP na desce ED2

Postup při vývoji aplikace

1. Vytvoření nového projektu v prostředí uScope
2. Napsání vlastního zdrojového textu
3. Přeložení zdrojového textu – vznikne soubor s příponou HEX
4. Nahrání přeloženého programu do procesoru pomocí programu Flip
5. Spuštění programu v přípravku

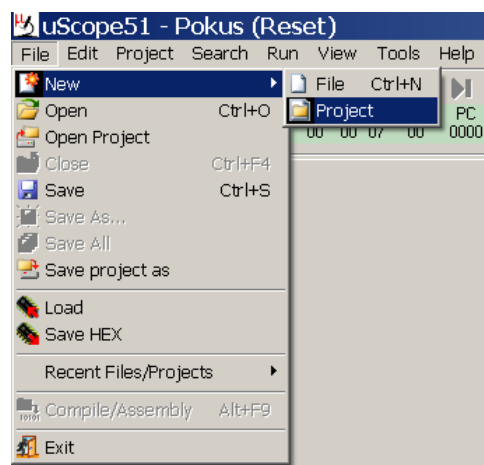
Návod na obsluhu programu uScope

Vytvoření nového projektu

V hlavním menu zvolte File – New – Project
Zvolte adresář, kam má být projekt uložen
Zobrazí se Project Manager, do kterého je možné přidat soubory, které patří do projektu.
Pokud ještě žádný takový soubor neexistuje, stiskněte OK

V hlavním menu zvolte File – New – File
Uložte nově vzniklý soubor do adresáře s projektem a jako příponu zvolte ASM

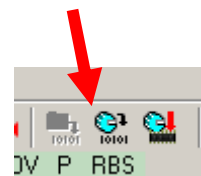
V této chvíli můžete začít psát zdrojový text.



Přeložení programu

Překlad programu lze spustit z menu Project – Build nebo pomocí ikony na liště rychlého spouštění.

Pokud ještě nebyl do projektu přidán žádný soubor, zobrazí se před překladem Project manager, do kterého je nutné pomocí tlačítka ADD přidat soubor se zdrojovým textem v Assembleru.



Návod na obsluhu programu Flip (verze 3.1.1)

Kontrola připojení desky

Pro připojení nahrávacího programu FLIP je nutné nejdříve na desce ED2 svorku X15 spojit propojkou (uzemnit signál PSEN) Signalizaci napájení desky ED2 signalizuje zelená dioda. Pokud dioda nesvítí je nutné přivést na desku napájecí napětí o hodnotě 9V. Jako komunikační rozhraní by měl být připojen kabel RS-232 nebo převodník USB/RS-232 k desce a počítači.

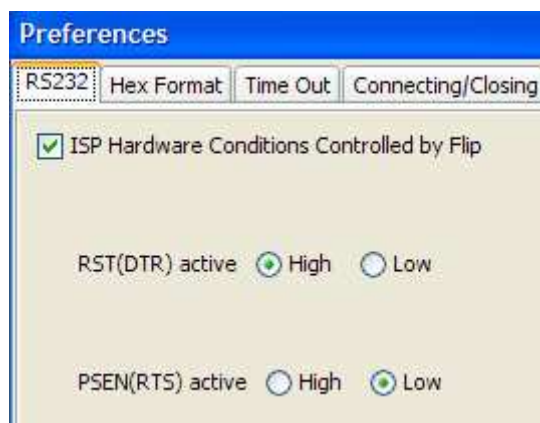


Výběr procesoru

Následně je nutné v ikonovém menu zvolit tlačítko zobrazující čip „*Select a Target device*“ a vybrat typ procesoru AT89C51RD2

Nastavení komunikace

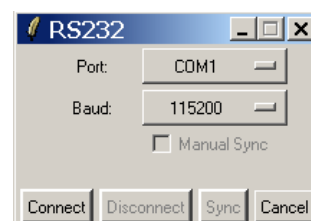
Aby mohl program Flip automaticky přepínat procesor do programovacího režimu musí být po spuštění utility FLIP v záložce „*Settings*“ vybrána volba „*Preferences*“. Po otevření dialogového okna bude zvolena záložka „*RS232*“. Po označení bodu s názvem „*ISP Hardware Conditions Controlled by Flip*“ musí být signály zvoleny do těchto úrovní: „*RST(DTR)*“ aktivní pozice „*High*“ a signál „*PSEN(RTS)*“ do aktivní pozice „*Low*“.



Pokud by bylo toto nastavení provedeno špatně bude program Flip při pokusu o komunikaci hlásit chybu Timeout.

Navázání komunikace

Dále bude zvoleno v ikonovém menu tlačítko symbolizující připojení „*Select a Communication medium*“ a zvolena volba RS-232. Po otevření záložky je zvolen komunikační port a rychlost připojení. Důležité je v tento moment vyresetování



mikroprocesoru tlačítkem „*Reset*“ na vývojové desce ED2. Nyní bude provedeno spojení tlačítkem „*Connect*“.

Výběr programu

Po propojení desky ED2 a počítače je v záložce „*File*“ vybrána volba „*Load Hex File*“. Je zvolen program, který bude do desky ED2 nahrán. Musí se jednat o již přeložený program (*.hex).

Nahrání programu do procesoru

Po výběru programu bude spuštěno tlačítkem „*Run*“ vymazání a nahrání programu do procesoru AT89C51ED2.

Spuštění aplikace

Program nahraný v desce ED2 bude spuštěn po potvrzení tlačítka „*Start application*“.